Using the Domain/PCI Connection

Order No. 007460-A00

Apollo Computer Inc.
330 Billerica Road
Chelmsford, MA 01824

Contents

Part 1. Getting Started

Chapter 1 Introduction to the Domain/PCI Connection

Your Connection to the Domain System 1-2
The Domain/PCI Connection Product Family 1-2
Key Terms and Conventions 1-5
Domain Resources 1-6
Terminal Emulation 1-6

Chapter 2 Installing and Starting the Domain/PCI Connection

Before You Begin 2-2
  What the User Should Read 2-2
  What the System Administrator Should Read 2-3
Installing the Domain/PCI Connection Software 2-3
  If Your PC Has a Fixed Disk 2-5
  If Your PC Has Diskette Drives Only 2-6
Starting the Domain/PCI Connection 2-9
  Invoking the Default Client Startup File 2-9

Invocation Options 2–11
What's Next? 2–12

Chapter 3 Making the Initial Domain/PCI Connection

Logging In 3–2
Connecting to Domain Directories 3–3
  Making an Initial Connection 3–3
  Making Additional Connections 3–6
  Specifying an Additional Number of Drives 3–7
Differences between DOS and Domain/OS Files 3–9
  Filename Extensions 3–11
  Filename Display Differences 3–12
  Pathname Translation 3–13
Displaying Domain/OS Filenames from DOS 3–13
Removing Directories 3–15
Logging Out 3–15
What's Next? 3–16

Part 2. User's Reference Guide

Chapter 4 Manipulating Files

The **append** and **assign** DOS Commands 4–2
  Accessing Files in Other Directories (**append**) 4–2
  Using Domain Files with Application Programs (**assign**) 4–4
Sharing Files with Domain Users 4–6
  What the File Type Conversion File Does 4–6
  Your Initial File Type Conversion File 4–7
  Locating the File Type Conversion File 4–9
  Creating Files on Domain from Your PC 4–9
  Storing a PC Application on Domain 4–10
  Which Conversion File Is Used? 4–12
  Converting Files without Extensions 4–12
  Accessing Files Created on Domain from Your PC 4–12
  DOS **Backup** and **Restore** Files 4–15
  Adding Comment Lines to the File Type Conversion File 4–16
Accessing Files 4–17
  Using a Text Editor 4–17
  Using a Word Processing Application 4–18
Copying Files 4–19
Printing Files 4–20
  How the Offers File Works 4–20
  Print Command Method 4–22
  Copy Command Method 4–23
  Print and Copy Method Examples 4–24
  Using Domain Printers 4–25
  Printing from Application Programs 4–26
What's Next? 4–28

Chapter 5 The Domain/PCI net Command Reference

net Command by Task 5–2
net Commands by Name 5–4
  net connect (directory) 5–5
  net connect (printer) 5–9
  net connect (list connections) 5–11
  net continue 5–13
  net disconnect (directory) 5–15
  net disconnect (printer) 5–17
  net help 5–18
  net login 5–20
  net logout 5–22
  net offers 5–24
  net pause 5–26
  net start 5–28

net use (directory) 5–30
net use (printer) 5–33
net use (disconnect from directory) 5–35
net use (disconnect from printer) 5–36
net use (list connections) 5–37

Part 3. DTERM User's Guide

Chapter 6 Getting Started with DTERM

Starting DTERM 6–2
  If You Use the **–on** or **–login** Options 6–4
  Account Protection 6–5
  Naming Directories 6–5
  Startup Scripts 6–5
The DTERM Display 6–6
  The Status Line 6–7
Using the Hot Key Feature to Switch between Domain/OS and DOS 6–9
Stopping DTERM 6–10
  The **logout** Command Method 6–10
  The <SHIFT><ESC> Method 6–11
Starting Up DTERM with a Specific Program 6–11

Chapter 7 Using the Domain System with DTERM

Using Shell Commands 7–1
Starting a Shell from Domain/OS 7–2
Using Pathnames 7–3
Using DPSS/Mail 7–3
Displaying Filenames and Directories 7–4
Invoking the **pci_assist** Command 7–5
Getting Access to "Invisible" Files 7–6
Creating and Using Script Files 7–7
  Creating Script Files 7–7
  Using Script Files 7–8

Chapter 8 Using VT100 Applications with DTERM

Invoking an Application 8–2
Character Attribute Support 8–2
Graphics Character Set Support 8–3
PC Equivalents for VT100 Keyboard Keys 8–4
Setting Screen Characteristics 8–5
  Using SET–UP A Mode 8–5
  Using SET–UP B Mode 8–8

Part 4. System Administrator's Guide

Chapter 9 Managing the Domain/PCI Connection Server

Before You Begin 9–2
Setting Up the Offers File 9–3
  Setting Up Domain Printer Offers 9–4
  Setting Up Directory Offers (Versions 1.0 and 2.0) 9–9
  The Initial Offers File 9–10
  Creating New Offer Definitions 9–10
Configuring the Server 9–11
Starting Domain/PCI Server for Direct Connection Products 9–13
Starting the Domain/PCI Server for Domain/PCI–1 9–14

Using Serial Expansion Units 9-15
The Server Processes Domain/PCI Creates 9-16
Stopping the Server 9-18

Chapter 10 Using Permission Rights

The Server's Subject Identifier (SID) 10-2
The 'node_data Directory 10-3
The Error Log File 10-4
Locked Objects when Permission Rights are Correct . 10-6
How DOS Responds to Permission Violations 10-7

Chapter 11 System Administration Command Reference

bldt 11-2
ctnode 11-4
lcnode 11-5
netstat 11-6
probenet 11-8
uctnode 11-10

Appendixes

Appendix A Messages and Process Status Errors

Messages A-2
Process Status Errors A-13
Server Process Status A-13
DTERM Process Status A-14

Appendix B Command Summary

Appendix C The Domain/PCI Connection Client Program Diskette

Executable Files in \dpci C-2
Help Files in \dpci C-3
Additional Files in \dpci C-4
Files in Top-Level or Root Directory C-4
Files on Domain/PCI-Enet Diagnostic Software Diskette C-4

Appendix D Customizing the Client Startup and \autoexec.bat Files

Customizing the Domain/PCI Client Startup File D-2
Editing the **net start** Command D-3
Editing the **net login** Command D-5
Editing the **net connect** Commands D-6
Customizing Your PC's **autoexec.bat** File D-7

Appendix E Updating the Client Software from an Install Area

Setting Up the Install Area E-2
Updating the Client Software E-3

Index

Figures

1-1 The Domain/PCI Connection Product Family 1-4

2-1 The Server and Client Programs of the Domain/PCI Connection 2-4

4-1 Converting a PC Text File to a Domain Text File 4-10

4-2 Preventing Conversion from a Domain File to a PC 4-13

6-1 DTERM Display before Login 6-3

6-2 DTERM Display after Login 6-6

8-1 VT100 SET-UP A Mode 8-6

8-2 VT100 SET-UP B Mode 8-8

Tables

1-1 Product Name and Value of *product* 1-5

3-1 Differences between DOS and Domain/OS Filenames 3-10

4-1 How the Offers File and User Input Interact . 4-21

5-1 The net Commands 5-3

8-1 VT100 Graphics Character Equivalences 8-3

8-2 VT100 Keys on the PC 8-4

Chapter 1

Introduction to the Domain/PCI Connection

The **Domain®/PCI™** (Domain/PC Interconnect) **Connection** lets you connect your personal computer (PC) to the Domain system. This powerful interconnection enables you to share resources, such as files and printers, with the Domain system and to use Domain services like electronic mail.

This chapter introduces concepts that will help you get the most from your Domain/PCI Connection. You'll learn about

- The connection between your PC and the Domain system
- The Domain/PCI products family
- Domain system resources
- Key terms and conventions
- Terminal emulation

Your Connection to the Domain System

The Domain system consists of computers, called **nodes**, linked to gether using a local area network. Any node on the network ca use the resources (such as files and printers) of any other node i the network. The Domain/PCI Connection lets your PC acces these same resources.

When you use the Domain/PCI Connection, the entire Domain/O file system becomes a **virtual drive** to your PC. This virtual drive i your connection to the Domain system, its resources, and its services.

The interactions between your PC and the Domain system are handled by a node on the Domain system called the host or **server node.**

The Domain/PCI Connection Product Family

The Domain/PCI Connection provides a variety of interconnection solutions, such as

- Domain/PCI-Enet
- Domain/PCI-Ring
- Domain/PCI-1
- Domain/PCI-8

This section introduces each member of the family.

The **Domain/PCI-Enet** Connection lets you connect your PC directly to an Ethernet network through an 802.3-compatible Ethernet adapter board. Domain/PCI-Enet provides full access to Domain/PCI Connection capabilities by connecting your PC to an Ethernet network on which a server node is running.

The **Domain/PCI-Ring** Connection lets you connect your PC directly to the Domain Token Ring network through the **Domain/PCI-Ring** network adapter board. Domain/PCI-Ring provides full access to Domain/PCI Connection capabilities by connecting your PC to an Domain Token Ring network on which a server node is running.

The **Domain/PCI-1** connection lets you connect your PC to a Domain node's Serial Input/Output (SIO) port.

The **Domain/PCI-8** connection lets you connect up to eight PCs to a Domain node's Serial Input/Output (SIO) port through a Domain/PCI-8 distribution panel.

Figure 1-1 shows how the Domain/PCI Connection product family connects PCs to the Domain system. The Domain/PCI Connection enables you to access the resources of any node on the Domain system from your PC. However, a user who is logged in at a node on the Domain system cannot access the resources of your PC.

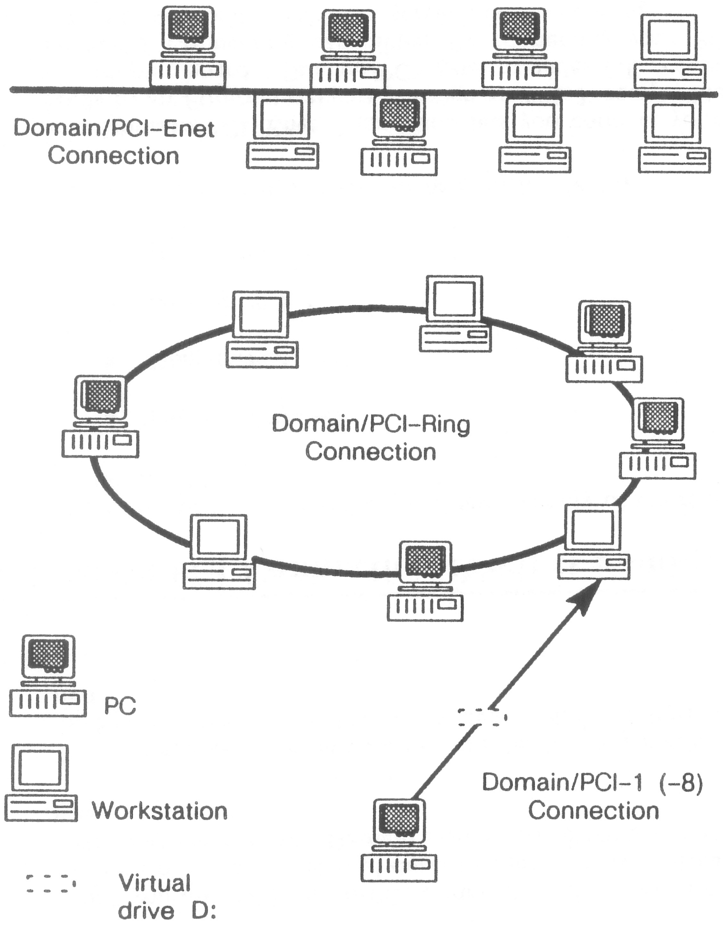


Figure 1-1. The Domain/PCI Connection Product Family

Key Terms and Conventions

We define the following key terms and conventions that are used throughout this manual:

A **direct connection product**, such as Domain/PCI-Ring, enables a PC to connect directly to a network.

An **SIO connection product**, such as Domain/PCI-1, enables a PC to connect to a network via a node's Serial Input/Output (SIO) port.

Domain/PCI Connection is a generic term that describes all Domain/PCI products.

Since all of the Domain/PCI products share the same capabilities, we use the variable name *product* in place of the actual name of the Domain/PCI Connection product you're using. For example, if you're installing the Domain/PCI Ring software, you would type **ring** in place of *product* in the following command:

```
pcinstl dpciproduct C:
```

```
pcinstl dpciring C:
```

Table 1-1 shows you which category your Domain/PCI Connection product is in and also lists the values of *product*.

Table 1-1. Product Name and Value of product

Product Name	Product Category	Value of *product*
PCI-Ring	Direct Connection	**ring**
PCI-Enet	Direct Connection	**enet**
PCI-1	SIO Connection	**1**
PCI-8	SIO Connection	**8**

Domain Resources

A **resource** may be a directory, a file, or a device, such as a printer.

The directories and files on your PC's diskettes and fixed drive, and the devices physically connected to your PC are known as **local resources**.

The Domain system resources available to your PC through the Domain/PCI Connection are known as **remote resources**. Most of this book discusses how to connect to the Domain system and use Domain system resources from your PC.

Once you connect to a remote resource you can use that resource with applications and commands in the same way you would use a local resource. For example, you can use the DOS **dir** command to look at the files contained in a directory on a Domain node. Chapters 3 through 5 provide this information.

Terminal Emulation

The Domain/PCI Connection provides a VT100* terminal emulation program called **DTERM**. When running DTERM, your PC can emulate the behavior of a VT100 terminal (without graphics or windowing capabilities).

When you run DTERM, you can use the services available to Domain/OS, including the Aegis, BSD, or SysV shell commands, DPSS/Mail™, and applications designed to run on a VT100 terminal. Chapters 6 through 8 tell how to use DTERM.

◆

---

*VT100 is a registered trademark of Digital Equipment Corporation.

Chapter 2

Installing and Starting the Domain/PCI Connection

The Domain/PCI Connection has two software components that you need to install. One component is for the server node on the Domain system where you connect your PC; the other component is for your PC. We refer to the software you install on your PC as **client software** and refer to the software the system administrator installs on the server node as **server software.**

In this chapter we explain how to:

- Install the Domain/PCI Connection client software
- Start the Domain/PCI Connection client program

Before You Begin

The following sections list which manuals you and the system administrator should read prior to reading this manual. Typically, any task that can be done from the PC is your (the user's) responsibility and any task that can done from the server node is the responsibility of the system administrator. To help you delegate these tasks, you should perform all the tasks described in the *Getting Started*, *User's Guide*, and *DTERM User's Guide* parts of this manual. The system administrator should perform all tasks described in the *System Administrator's Guide* of this manual.

What the User Should Read

Be sure you've read the following documents (in the order in which they are listed) completely *before* proceeding with this chapter. This is important because the Domain/PCI server software and PC hardware installation information is not covered in this manual:

- *Domain/PCI Connection Release Document*, which discusses how to install the software on the server and describes client and server software bugs and limitation information.

- *Planning for and Installing the Domain/PCI Connection*, which describes how to install and run diagnostic tests on product-specific Domain/PCI Connection hardware.

- Any documentation, such as installation guides and addenda, that comes with connection hardware products from vendors other than Apollo Computer, such as the 3Com EtherLink II adapter board.

What the System Administrator Should Read

You should read all of the documents listed in the section "What the User Should Read," in addition to Chapter 9 of this manual, which describes how to set up and start the server software. You must perform the server software installation steps described in the release document and the server management procedures in Chapter 9 of this manual before the user can install the client software and start the Domain/PCI Connection on the PC.

Installing the Domain/PCI Connection Software

The Domain/PCI Connection consists of a **server** program (or process) and a **client** program. When you are using the Domain/PCI Connection, these programs exchange information. The server program runs on the server node that your PC is logically connected to, and the client program runs on your PC.

Figure 2–1 illustrates the relationship between the Domain/PCI Connection server and client programs.

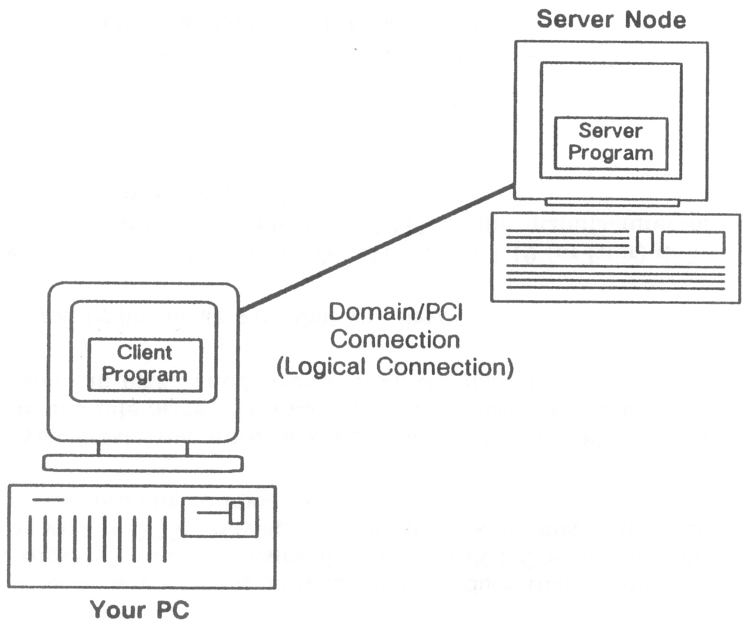


Figure 2-1. The Server and Client Programs of the Domain/PCI Connection

You can install the Domain/PCI Connection client software on a fixed disk or make an additional copy of the Domain/PCI Connection client software if you're using a floppy diskette drive. Refer to the diskette containing the Domain/PCI Connection client software as the **master** diskette. Some products may include more than one client software diskette.

NOTE: The Domain/PCI Connection master diskette is a 360K (double-sided, double-density) diskette. Your PC must have at least one diskette drive capable of reading a 360K diskette.

If Your PC Has a Fixed Disk

This section describes how to install the Domain/PCI Connection client software on your PC's fixed disk.

Before you begin, make sure that

- DOS is installed on your fixed disk.
- Your fixed disk has at least 160K of free space.
- You have your Domain/PCI Connection master diskette.

Follow these steps to install the Domain/PCI Connection client software on your PC's fixed disk:

1. Place a write-protect tab on your Domain/PCI Connection master diskette so that it cannot be overwritten accidentally.
2. Insert the Domain/PCI Connection master diskette in drive A.
3. Change the default drive to drive A with this command and press <RET>:

   ```
   C:> a:
   ```

4. Start the installation by entering the following command and pressing <RET>:

   pcinstl dpci*product*

   where *product* is the name of the product you're installing (as listed in Table 1-1).

   A message similar to this follows:

   ```
   About to copy dpciproduct Rel n.n client software for
   dpciproduct to C:
   Strike a key when ready . . .
   ```

5. Press any key to continue, or press <CTRL> C to stop the installation and return to the DOS prompt.

The installation checks for the existence of the **dpci** directory on your fixed disk; this directory is where the Domain/PCI Connection software will be installed.

If the **dpci** directory does not exist, the procedure creates the directory.

When the procedure is completed, the DOS prompt is displayed:

```
A:>
```

6. Change your default drive to drive C, and remove the Domain/PCI Connection master diskette from drive A.

You are now ready to start the Domain/PCI Connection program. Go to the next section in this chapter, "Starting the Domain/PCI Connection."

If Your PC Has Diskette Drives Only

This section describes the installation procedure that produces a **work diskette,** which contains the Domain/PCI Connection software as well as DOS. Having DOS on the diskette means that you can use the diskette to boot your PC.

After you create the work diskette, put the Domain/PCI master diskette in a safe place. This way, if the work diskette is ever damaged, you can make another copy from the master.

Before you begin, make sure that you have

- An MS-DOS* (or PC-DOS*) version 3.10 (or later) diskette, which is supplied by IBM* or the manufacturer of your IBM-compatible PC
- One blank 360K double-sided, double-density, 5-1/4-inch diskette; this is your work diskette
- Your Domain/PCI Connection master diskette

Creating a Work Diskette

Follow these steps to create your Domain/PCI Connection work diskette, which will contain both the Domain/PCI Connection software and DOS:

1. Ensure that there is a write-protect tab on your Domain/PCI Connection master diskette so that it cannot be overwritten accidentally.

2. Insert your DOS diskette in drive A and boot your PC.

3. Format the work diskette by first typing the following at the A:> prompt:

```
A:>  format b: /s
```

4a. *If your PC has two diskette drives*, insert the work diskette in drive B (the other drive).

4b. *If your PC has a single drive*, remove your DOS diskette from drive A and insert the work diskette in drive A.

5. Press <RET> and follow the messages on your screen.

When your PC finishes formatting, it displays this prompt:

```
Format another (Y/N)?
```

6. Type **n** (for no) at the prompt:

```
Format another (Y/N)? n
```

The work diskette is now formatted and contains DOS.

7a. *If your PC has two diskette drives*, remove your DOS diskette from drive A. Leave the work diskette in drive B and continue with the procedure in the next section, "Copying the Domain/PCI Connection Software to Your Work Diskette."

---

*MS-DOS is a registered trademark of Microsoft Corp.

*PC-DOS and IBM are registered trademarks of International Business Machines Corporation.

7b. *If your PC has a single drive*, remove the work diskette from the drive and continue with the procedure in the next section, "Copying the Domain/PCI Connection Software to Your Work Diskette."

Copying the Domain/PCI Connection Software to Your Work Diskette

Follow these steps to copy your Domain/PCI Connection software to the work diskette:

1. Insert the Domain/PCI Connection master diskette in drive A.

2a. *If your PC has two floppy drives*, insert the work diskette in drive B.

2b. *If your PC has a single drive*, wait for the message to tell you when to change diskettes.

3. Start the installation by entering the following command and pressing <RET>:

   pcinstl dpci*product* b:

   where *product* is the name of the product you're installing (as listed in Table 1-1).

   A message similar to this follows:

```
About to copy dpciproduct Rel n.n client software for
dpciproduct to drive:
Strike a key when ready . . .
```

   NOTE: If your PC has a single diskette drive, you are prompted to change disks during the copy from the master to the work diskette.

4. Press any key to continue, or press <CTRL> C to stop the installation and return to the DOS prompt.

   The work diskette now contains *both* the Domain/PCI Connection software and DOS.

5a. *If your PC has two diskette drives*, remove the master diskettes from drive A. Insert the work diskette in drive A.

5b. *If your PC has a single drive*, leave the work diskette in drive A.

You are now ready to start the Domain/PCI Connection.

Starting the Domain/PCI Connection

This section describes how to start the client program, which you installed onto your hard disk or copied to a work diskette.

Before you start the client program, you or your system administrator must make sure that the Domain/PCI Connection server software has been installed and is running on the server node. Your system administrator should refer to Chapter 9, which describes how to start the Domain/PCI Connection server program.

> NOTE: The Domain/PCI-1 Connection lets you use a modem to connect your PC to the server node (described in *Planning for and Installing the Domain/PCI Connection*). If you are connecting through a modem, notify your system administrator of the modem's baud rate before you start the Domain/PCI Connection.

Invoking the Default Client Startup File

The **default client startup file** is a batch file that we provide with all products in the Domain/PCI Connection family. When invoked, this file prompts you to log in and automatically makes two initial connections to the Domain/OS system. One connection establishes drive D as the **top level** or **root directory** of the Domain/OS system and the other connects the device drive named **lpt2** to the default Domain printer.

Follow these steps to start the Domain/PCI Connection client program on a PC using the default client startup file, **\dpci\dpci*product*.bat**, where *product* is the name of the Domain/PCI Connection product you're using.

1. Make sure that the Domain/PCI Connection server is running on the server node. Check with your system administrator to be sure.

2. Make sure that you are running DOS 3.10 (or later) on your PC. To verify that you are, type:

   ```
   ver
   ```

   Do not continue this procedure until your PC is running DOS 3.10 (or later).

3. *If you are using a PC with diskette drives only*, insert your Domain/PCI Connection work diskette in drive A, and boot your PC. (The procedure for creating the work diskette is described earlier in this chapter.)

4. Start the Domain/PCI Connection client program by changing the directory to **\dpci** as shown below

   ```
   cd \dpci
   ```

5a. If you want to start the default client startup file of a Domain/PCI direct connection product (as listed in Table 1-1), enter the following invocation command and press <RET>

```
dpciproduct server_id [-option]
```

where *product* is the name of the product you're using. The *server_id* is the hexadecimal node ID of the server node you want to connect to. If you do not know the ID of the server node, ask your system administrator. (Note to system administrator: Both the server and the PC must be on the same network. Refer to the **bldt** command in Chapter 11 for details.) The *option* used with this invocation command is described in the "Invocation Options" section later in this chapter.

For example, **dpciring 18dfc -retries 25**, starts Domain/PCI-Ring on a server with a node ID of 18dfc.

5b. If you want to start the default client startup file of a Domain/PCI SIO connection product (as listed in Table 1-1), enter the following invocation command and press <RET>

```
dpci1 [-options]
```

The *options* used with this invocation command are described in the following section, "Invocation Options."

For example, **dpci1 -line1 -baud 4800 -retries 25**

6. When the default client startup file displays the login prompt, you're ready to log in. Go to Chapter 3.

Invocation Options

The Domain/PCI direct connection products have one invocation option. The Domain/PCI SIO connection products have three invocation options. You can include one or more of these options to modify the invocation command.

Direct Connection Products Options

-retries *zz*	Specifies the number of times the PC should try to execute a command again after encountering an error. The default number is 60. The maximum number of retries is 250. If you are on a busy network, you may want to use a number greater than the default to ensure that you stay connected to the network. For example, **dpciring 18dfc -retries 95**

SIO Connection Products Options

-line *n*	Indicates the number of the asynchronous communication adapter connector where the cable for the Domain/PCI Connection server is plugged in; this number *n* can be 1 or 2. **1** indicates COM1 (PRIMARY or LINE 1) and **2** indicates COM2 (ALTERNATE or LINE 2). For example, **dpci1 -line 1**

-baud *yyyy*	Specifies the baud rate, where *yyyy* is the rate. The value of *yyyy* can be 300, 600, 1200, 2400, or 4800. For example, **dpci1 -baud 4800** You need to specify this option only if you have established a baud rate on the server node that is different from 9600. (System administrators: see Chapter 9.)
-retries *zz*	Specifies the number of times the PC should try to re-execute a command after encountering an error. The maximum number of retries is 250. The default number is 10. If you are on a busy network, you may want to use a number greater than the default to ensure that you stay connected to the network. For example, **dpci1 -retries 25**

What's Next?

In this chapter, you learned how to install and start the Domain/PCI Connection. You also learned the invocation commands for direct connection and SIO connection products.

In the next chapter, you'll learn how to log in and see how you can access the Domain system from your PC. You'll also learn about the differences between the DOS and Domain file conventions.

After you've read through Chapter 3 and performed the exercises, you may want to customize the default client startup file and the **autoexec.bat** files. If you customize these files, your PC can automatically start the Domain/PCI Connection when you turn it on. It also allows you to log in and makes your customized connections automatically. How to customize these files is described in Appendix D.

◆

Chapter 3

Making the Initial Domain/PCI Connection

Once you've started the Domain/PCI Connection, you can log in and let the default client startup file make your initial connections to Domain resources. The initial connections will automatically connect your PC to the top level or root directory of the Domain system and connect lpt2 to the default Domain printer.

This chapter shows you how to log in, access the initial connection, and make additional connections. It takes you for a quick tour through some of the basic features and benefits of the Domain/PCI Connection. Chapter 5 contains full descriptions of all the commands presented in this chapter.

After completing this chapter, you'll be able to

- Log in
- Connect to Domain directories
- Understand differences in filenaming conventions between DOS and Domain/OS
- Display Domain filenames from DOS
- Remove directories
- Log out

Logging In

Once you've executed the invocation command (described in Chapter 2), you're ready to log in. Your system administrator should have set up an account for you (with a user name and a password) so that you can log in. Before proceeding, be sure that you have an account.

Logging in enables you to tell the server who you are. The server needs to know your identity before it can grant access to files and directories. This is because Domain files have permission rights. Without the proper permission rights associated with your login name and password (called **login account**), you won't be able to access protected files and directories.

To log into the Domain/PCI Connection, follow these steps:

1. At the login prompt, enter your user name and press <RET>:

   ```
   Please log in:
   ```

   The following prompt displays:

   ```
   Password:
   ```

2. Enter your password and press <RET>. Note that the screen displays a dot for each character you enter. This prevents others from seeing your password as you enter it.

   You may need to wait about 30 seconds before the login command successfully completes.

   When the command completes, you'll see this message:

   ```
   Command completed successfully.
   ```

   You are now logged in.

Your default client startup file will make an initial connection to the Domain system (D:) and will make a connection to the Domain default printer (**lpt2:**). The next sections describe how to access the initial connection and how to make additional connections.

Connecting to Domain Directories

To connect your PC to a Domain directory, you need to associate the name of a drive device with that directory. You can use the **net connect** command (or the **net use** command for versions 1.0 and 2.0 of Domain/PCI only) to make connections to the Domain system. This section shows you how the default client startup file uses the **net connect** command to make your initial connection, and how you can use this command to make additional connections.

Making an Initial Connection

The default client startup file contains a command that makes an initial connection to the top–level or root directory of the Domain file hierarchy. This command is

```
net connect d: //
```

After you logged in, the default client startup file executed the initial connection command and connected your PC to the Domain system's top–level directory.

Then, a second command in the default client startup file connected **lpt2** to the default Domain printer. This command is

```
net connect lpt2: print
```

(However, since printing a file is a subject of Chapter 4, we'll delay discussion of it in this chapter.)

If both connections were successful, each was followed by the message

```
Command completed successfully
```

The DOS prompt (C:>) is displayed after making the connection to the default printer. You now have two initial connections to Domain.

The following steps show you how to access Domain through the connection to the top-level directory on drive D:

1. Enter the name of the drive that connects your PC to the top-level directory on the Domain system and press <RET>.

   ```
   C:> d:
   ```

   The D: prompt is displayed. This is your connection to the top-level Domain directory, which allows access to all the nodes in the network. From the top-level directory, you can access other directories and files below by using DOS commands. However, for this exercise, use only the DOS commands used in the procedures.

   You might be tempted to use the DOS **dir** command to see a directory listing. Before you do, be aware that executing the **dir** command from the top level of a network displays a list of all the nodes in the network. If your network contains many nodes, executing the **dir** command from the top level can take a long time.

2. At the D: prompt, change the directory to a subdirectory and press <RET>. For example,

   ```
   D:> cd \wayside\plans
   ```

   If the connection is successful (and if your **prompt** command is configured as **prompt pg** in your **\autoexec.bat** file), you'll see the D: prompt followed by the pathname of the subdirectory you specified in this step. Here's what our example gives us:

   ```
   D:\WAYSIDE\PLANS>
   ```

Now that you've connected a drive to a Domain directory, you can use DOS commands to list the directory and to copy, look at, rename, and delete files in that directory. The following sections show you how to display the contents of a directory, how to read a file, and which other DOS commands you can use.

How Do I Display the Contents of a Directory?

Let's say you have successfully connected drive D to the Domain top-level directory and have accessed the subdirectory **\wayside\plans**. To list the contents of the **plans** directory, type this DOS command:

```
D:\WAYSIDE\PLANS> dir


Volume in drive D is WAYSIDE
Directory of D:\WAYSIDE\PLANS


README    5-29-88         3:22p

.
.
.
 24 File(s) 1138688 bytes free
```

When you make a Domain/PCI connection, the **volume label**, (in this case, **WAYSIDE**) is the node name of the current directory. The **bytes free** is the free space on the node listed as the volume label. If the current directory is the network root directory, the double slashes (//) display as the volume label and the bytes free are the server node's disk free space.

How Do I Read a File?

To read a text file in a directory, use the DOS **type** command. For example, let's say that you want to read a file in the **plans** subdirectory called **readme**. To read that file, type this command:

```
type readme
```

To stop the file from scrolling, press <CTRL> S. To resume scrolling, press <CTRL> S again.

Which Other DOS Commands Can I Use?

You can use all DOS commands except those that read or write directly to the disk. (For descriptions of DOS commands, consult the DOS User's Guide and the DOS Reference Manual.)

You cannot use the following DOS commands with Domain resources:

CHKDSK FDISK SYS LABEL RECOVER

DISKCOMP SUBST JOIN FORMAT DISKCOPY

If you try to use any of these commands with a Domain resource, this message is displayed:

```
Cannot COMMAND to a network drive
```

where *COMMAND* is the name of the command you used.

Making Additional Connections

After you've made the initial connections and have accessed other directories, you may find that there are several specific directories that you normally need to access. You could wend your way from the top-level connection to other directories, but that involves remembering several pathnames and typing long pathnames. Or, you could simply connect additional drives to other Domain directories. This section show you how to make those additional connections.

To connect to additional directories, perform these steps:

1. Return to the root level of the physical drive.
2. At the DOS prompt (C:>), enter the Domain drive and the pathname of the directory you want to access. For example,

   ```
   C:> net connect e: \\wayside\plans
   ```

3. Press <RET>. The message,

   ```
   Command completed successfully.

   C:>
   ```

   and the DOS prompt are displayed.

   You have made a new connection to drive E:. The root of drive E is the Domain directory **//wayside/plans**. The root of drive D is the Domain directory //. To list your connections, perform Step 4.

4. At the DOS prompt, type **net connect** and press <RET>. A list of your connections is displayed. For example,

```
DOMAIN (R)/PCI(TM) CONNECT x.xx May x, 19xx, x:xx:xx am

Default node id:  18DFC
Default username:  DAVID

Local Name Remote Name

D:                        //18DFC/DAVID
Path is:                  //

E:                        //18DFC/DAVID
Path is:                  \\WAYSIDE\DAVID

LPT2:                     //18DFC/PRINT

Command completed successfully.
```

Specifying an Additional Number of Drives

You may want to add the Domain/PCI **lastdrive** command to your PC's configuration file, **\config.sys**, to increase the number of drives available for additional connections. This special command allows you to increase the number of virtual drives beyond the standard DOS limit of five drives.

Your PC's configuration file, **\config.sys**, contains commands that DOS executes at startup. These commands enable you to change your system's default configuration settings to meet your needs. By default, DOS enables you to make connections to drives A through E. However, in order to make additional connections beyond the drive letter E, you'll need to add the **lastdrive** command. Refer to your DOS Reference Manual for additional information about devices and the **\config.sys** file.

What Is a Drive Device?

A drive can be a physical device, such as a fixed disk (C:) or floppy drive (A:), or it can be a virtual drive (D:), such as a connection to Domain. The name of a drive is always a single letter. The name may refer to a physical drive, like A or C, or it may be a virtual drive, such as D or F.

By adding the **lastdrive** command to the **\config.sys** file, you may have a total of 26 drives with drive names A through Z. You may not have more than ten connections to different directories at one time.

How Do I Increase the Number of Virtual Drives?

To increase the number of virtual drives, you need to edit or create (if the file doesn't exist) the PC's **\config.sys** file to add the **lastdrive** command. Using a text editor, such as **edlin**, add the **lastdrive** command to indicate the name of the last drive you want. For example, in the last line of this sample **\config.sys** file, we specify **z** as the last drive, for a total of 26 drives:

```
file=20
BUFFERS=6
lastdrive=z
```

If the **lastdrive** command is already in the configuration file, you can edit the command to meet your needs. For example, to change from 26 drives to 10 drives, you would edit the command **lastdrive=**z to read **lastdrive=**j.

After editing **\config.sys**, you'll need to restart (boot) your PC so that the PC can read the new value that you've entered.

Differences between DOS and Domain/OS Files

Now that you've made a few connections to the Domain/PCI Connection, you're ready to manipulate (access, copy, and print) files. But first, you need to understand the differences between DOS and Domain files before you can manipulate a file.

The rest of this chapter describes the differences between DOS and Domain filenames and shows you how to resolve the differences when necessary. Chapter 4 shows you how to manipulate files.

The Domain/PCI Connection acts as a bridge between two different operating systems, DOS and Domain/OS. These operating systems use somewhat different conventions for naming files and directories. This section describes the different conventions used by both operating systems.

Table 3-1 summarizes the differences between DOS and Domain/OS files.

Table 3-1. Differences between DOS and Domain/OS Filenames

Property	DOS	Domain
Filename Length	1-8 characters	1-32 characters (pre-SR10)
		1-255 characters (SR10)
Filename Extensions	1-3 characters (always begin with period)	Not explicit
Naming Conventions	1 period allowed in filename	Multiple periods allowed
Filename Display	Displays filenames in uppercase	Lowercase filenames displayed
	Mixed case or uppercase filenames are not displayed	Mixed case or uppercase filenames are displayed (SR10)
Pathname Translation	\root of directory	//root of directory //root of node
Filename Characters	A-Z a-z 0-9 $ % ' - @ { } ~ ‘ ! # (blank space not displayed)	A-Z a-z 0-9 $ % ' - @ { } ~ ‘ ! # blank space
	(not displayed)	* + , ; < > = ? : [ ] | ” \

As you can see from Table 3-1, there are many differences between Domain/OS and DOS filenames:

- Domain/OS filenames can be longer.
- Domain/OS filenames can have multiple periods.
- Domain/OS filenames can contain characters not valid in DOS filenames.
- Domain/OS filenames can be mixed case or uppercase.

The sections that follow explain:

- The differences between the DOS and Domain/OS filename extensions
- How the Domain/PCI Connection displays DOS and Domain/OS filenames
- The differences between the DOS and Domain/OS pathname separator

Filename Extensions

A DOS filename may have an extension. An extension always begins with a period and contains one to three characters. For example,

```
CONFIG.SYS
```

If a DOS filename does not have an extension, the filename does not include a period; for example,

```
STATUS
```

A Domain filename does not have an explicitly defined extension. While a Domain filename may end with extension characters such as **.exe**, **.pas**, or **.c**, the last characters in the filename are not restricted. These are some examples of Domain filenames:

```
test.bin      test.bin.bak    test.c
test.c.bin    test.tuesday    test.tuesday.last
```

In a DOS filename, the only place that a period occurs is with an extension. In a Domain filename, a period can occur anywhere, and may occur more than once.

Filename Display Differences

Because of the differences between DOS and Domain filename conventions, the Domain/PCI Connection may change how some filenames are displayed when you

- Look at a DOS file or directory from DTERM
- Look at a Domain/OS file or directory from DOS
- Copy a DOS file to the Domain/OS system
- Copy a Domain/OS file to the DOS system

The examples below illustrate how the Domain/PCI Connection displays filenames:

As displayed on Domain:	**As displayed on DOS:**
`1example`	1EXAMPLE
`_read_.m`	_READ_ M
`:up:case`(pre-SR10)	(not displayed)
`UpCase` (SR10)	(not displayed)

Note that while the Domain system *displays* characters in lowercase (also mixed case and uppercase at SR10), DOS *displays* them in uppercase.

Also note that Domain files with mixed case characters (:up:case or UpCase) are not displayed on DOS.

Connecting to Directories with Case Differences

Since DOS can't display files with mixed case or uppercase, you can't make a connection to Domain directories that have mixed case or uppercase names. For example, if you attempt to connect to a Domain directory named **Big**, which is in all uppercase or mixed case, as in

net connect w: \\wayside\david\Big

you'll get an error message telling you that the path isn't found.

The section, "Displaying Domain/OS Filenames from DOS" provides solutions to this problem.

Pathname Translation

Within a pathname, the DOS "\" pathname separator is equivalent to the Domain "/" separator. However, at the beginning of a pathname, the DOS "\" translates to the root of the drive device (\\), which is where the network drive connects to the Domain/OS file system.

For example, suppose the pathname of a Domain directory is **//node5/beth**. The following example illustrates three equivalent ways of getting to the **beth** directory under the **//node5/beth** directory:

If you use:	**Equivalent DOS pathname is:**
**net connect d: **	**cd d:\node5\beth**
net connect e: \\node5	**cd e:\beth**
net connect f: \\node5\beth	**cd f:\ or f:**

Displaying Domain/OS Filenames from DOS

DOS can recognize only filenames and directory names that have no more than eight characters and a three-character extension. This is referred to as the **8.3 character format**. For example, the file **readtome.txt** is in the 8.3 character format. Some files in the Domain system may be invisible to your PC while it's running DOS.

For example, in the following directory **//node5/david**,

```
Directory "/david:

alter       batch2.txt          ch2.fmt         com
desktop     dpciring            help            install
mb          pc_list             release_pci     sys
```

the subdirectory **release_pci** isn't displayed when you list this directory on your PC:

```
Directory of E:\

COM          <DIR>          8-24-88    2:48p
ALTER        <DIR>          8-25-88   10:53p
MB           <DIR>          6-12-88   10:07p
DESKTOP      <DIR>          8-26-88    5:07p
CH2       FMT      29547    7-29-88   10:24a
DPCIRING     <DIR>          8-26-88    2:37p
PC_LIST              1326   8-07-88    4:47p
INSTALL      <DIR>          6-10-88    4:17p
SYS          <DIR>          6-10-88    8:47p
BATCH2.TXT            864   8-19-88    9:22p
HELP         <DIR>          8-05-88   11:57p
```

You can display the filenames and directory names that DOS cannot display with the **pci_assist** utility while you are using DTERM. The **pci_assist** utility is described in Chapter 7.

To provide your PC access to these invisible Domain files and directories, you must either:

- Change the Domain names to adhere to the DOS 8.3 character format.

  or

- Invoke DTERM (refer to Chapter 6) and create links to the Domain names with names that DOS can display. For example, create a link from **release_pci** to **rel_pci**. (For information on creating links, see the command reference for the environment you're using: Aegis, BSD, or Sys V.)

If the object with the invisible name you want to reach on the Domain system is a directory, you can use the **net connect** command to connect a virtual drive to that directory. The pathname used in the **net connect** command can be any Domain/OS pathname. For example, to reach the Domain/OS directory **//node5/david/release_pci**, you could use the following **net connect** command:

```
net connect e: \\node5\david\release_pci
```

In this example **e:** now points to **//node5/david/release_pci**.

Removing Directories

If you attempt to remove a directory with the DOS command **rmdir** and you get the following error message: "Invalid path, not directory, or directory not empty," the directory may contain files or subdirectories that do not appear because of the DOS file format convention described in this chapter. Use one of the two procedures described above to remedy the problem.

Logging Out

Logging out of the Domain/PCI Connection prevents others from accessing your protected files and directories.

To log out, follow these steps:

1. From a DOS prompt, type the following:

   ```
   C:> net logout
   ```

2. Press <RET>. The following message is displayed:

   ```
   Command completed successfully.
   ```

The **net logout** command breaks all connections between your PC and the Domain system. You can see that all connections are broken if you type **net connect** at the DOS prompt and press <RET>.

To reconnect to a Domain resource, you must start the Domain PCI Connection and log in again.

What's Next?

In this chapter, you learned how to log in and access the Domain system. You learned how to make additional connections to directories you might use frequently and how to log out. You also learned the differences in filenaming conventions between DOS and Domain/OS and how they affect your access to files and directories.

In the next chapter, you'll learn about the Domain/PCI file conversion file, **pci_cvrt.dat**, how to create and access text files, and how to copy and print files.

* * *

Chapter 4

Manipulating Files

Chapter 3 gave you a quick tour of the Domain/PCI Connection's features. By now, you know how to log in, make connections to directories in the Domain system, and log out. You also have a good understanding of the differences between DOS and Domain/OS files.

In this chapter, you'll learn how to

- Use the **append** and **assign** DOS commands with the Domain/PCI Connection
- Set up and use the file type conversion file
- Access a file
- Copy a file
- Print a file

The append and assign DOS Commands

Before you manipulate files, you may want to use two DOS commands:

- **append**
- **assign**

The **append** command helps you set a search path for files in directories other than the working directory. The **assign** command lets you read and write to files on drives other than A or B when an application typically uses only those drives. These commands are described below.

Accessing Files in Other Directories (append)

Suppose you have several customer files in the directory **\sales\accounts** on the Domain system. You want to edit those files with a word processing program. You can list each file by typing **\sales\accounts\**_filename_ each time you want a file, but an easier way is to use the **append** command to set a search path to the directory.

Let's say drive D is connected to the top-level directory of the Domain system. To set a path to the directory **\sales\accounts** on drive D, type this **append** command:

```
append d:\sales\accounts
```

Now, you can access the files in the directory **\sales\accounts** simply by typing only the filenames; you don't have to type the drive and pathname. The **\sales\accounts** directory on drive D is searched automatically to locate the files.

The **append** command tells DOS to search certain directories for files with extensions other than **.exe**, **.com**, or **.bat**. (You may be familiar with the DOS **path** command, which tells DOS to search certain directories for external command files. The **path** command works only for executable files that have one of these extensions: **.exe**, **.com**, or **.bat**.)

Format

append [*d:\path*] [; [*d:\path*]]...

where *d:\path* is the drive name and complete pathname of the directory you want DOS to search. For example,

```
append d:\sales\accounts;d:\sales\regions\reps
```

You can specify more than one path to be searched by separating each pathname with a semicolon (;), as long as you don't exceed 128 characters in the command line. DOS searches the directories in left-to-right order.

You can also use **append** to display the names of the directories DOS should search and to cancel the previous **append** command. These uses of **append** are explained in sections that follow.

Canceling an append Command

To cancel the most recent **append** command, type the following:

```
append;
```

Replacing an append Command

Entering an **append** command cancels any previous search paths. To replace the most recent **append** command, simply type a new **append** command. For example, let's say that after using the files in the **\sales\regions** directory on drive D, you want to use the files in the **\sales\regions\reps** directory on drive D. You would type this command:

```
append d:\sales\regions\reps
```

DOS now searches the **\sales\regions\reps** directory rather than the **\sales\regions** directory.

Displaying Search Paths

To display the current paths that DOS will search, type the **append** command as follows:

```
append
```

When you do so, your PC displays a list of the current search paths.

For example, let's say you have typed the following command to establish two search paths:

```
append d:\sales\accounts;d:\sales\regions
```

When you enter the **append** command, your PC displays the search paths:

```
A>append
APPEND=D:\SALES\ACCOUNTS;D:\SALES\REGIONS
```

Using Domain Files with Application Programs (assign)

Some applications read and write only to files that are in specifi drives. For example, an application might require data files to b in drive B. To use such an application, you need to use th DOS **assign** command to map the actual drive where the data file are located to the drive that the application requires.

For example, let's say you have a word processing program tha requires data files to be on drive B. You want to edit a file calle **loan** in the Domain directory **\letters\july** on drive D.

To do this, first set the default drive to D:

```
cd d:\letters\july
```

Then use the **assign** command to equate drive D (where the file is) and drive B (where the word processing program expects the file to be):

```
assign b=d
```

Now you can start the word processing program and edit the file **loan**. The word processing program's disk requests are routed to:

```
D:\letters\july
```

If you try to use an application that requires files from a specific drive, and the drive doesn't contain the necessary files, an error message is displayed. The error message may indicate either that the drive is not ready or that the file does not exist.

Format

assign *y*=*z*

where *y* is the name of the drive required by the application, and *z* is the actual drive you want to use.

Sharing Files with Domain Users

This section is intended for those users who will be sharing files with Domain users. If you will be sharing files only with other PC users and not with Domain users, you can skip this section and go to the section called "Accessing Files."

The text files that you create on your PC are in **text file** format. The text files that you create on the Domain/OS system are in **uasc** format. Examples of text files include program source files (written in C, Pascal, etc.) and files that you create with a PC text editor such as **edlin** (a DOS utility).

The primary difference between text files and uasc files is the way that lines (records) in files are terminated. Each line in a PC text file is terminated with both a carriage-return character and a linefeed character; each line in a uasc file is terminated with a linefeed character.

What the File Type Conversion File Does

The DOS file system doesn't have specific file types. Thus, when sharing files between DOS and Domain, you need a file type conversion file to specify which file extensions are text files that should be converted to uasc files on Domain.

The server uses a file type conversion file, **pci_cvrt.dat** (located in **/sys/dpci**), to control which files get converted to uasc files whenever you create or copy a file from your PC on Domain. Conversely, the **pci_cvrt.dat** file controls some special cases in which you may not want uasc files to be converted to text files when the file created on Domain is copied or read back to your PC.

The file type conversion file contains two sections: extensions and directives. The **extensions**, such as **.doc** or **.txt**, enable you to control file conversion when you're creating text files on Domain from your PC. If the extension is in the conversion file, the shared file is converted to uasc. The **directives**, which begin with a > character, enable you to control whether or not file conversion occurs when files created on Domain are read from a PC. If a directive is present in the conversion file, the directive determines file conversion.

Without proper file conversion, you may not be able to read or execute shared files. The following sections describe how to control file conversion from both your PC to Domain and from Domain to your PC.

Your Initial File Type Conversion File

The **pci_cvrt.dat** file that comes with the Domain/PCI Connection contains a sampling of file extensions and directives that you may want to use when you are sharing files between your PC and Domain. However, the extensions are commented out, which means that file conversion will not occur until you remove the comment characters (;) from those extensions that you want to use in the conversion. If you do not want any file conversion to occur, do not remove any of the comment characters. We simply provide these extensions as examples of files that are often text files on the PC. You can add both filename extensions and directives to the **pci_cvrt.dat** file or delete extensions and directives from it.

The initial file looks similar to this:

```
;
; Line beginning with '>' control conversion when reading
; files from Domain to the PC.
;
; nfs type uids don't get converted because we can't tell
; whether they are ascii or binary.
;
>uid nfs_dir        ; directories via nfs
>uid nfs_gate       ; generic nfs gateways
>uid nfs_file       ; nfs files don't get converted
;
; These are the extensions that when copying from domain to
; the pc don't get converted even if the file is uasc.
; This is to support cross compilers/assemblers that run
; on domain and generate ascii object files.
;
>not obj
>not exe
>not q              ; lattice intermediate file
;
; Extensions which do get converted
; Remove comments or add your own extensions below
;
;asm
;bak
;bat
;c
;dat
;doc
;ftn
;h
;lst
;pas
;txt
;map
;$$$ ; This is used by some editors as the temporary file
;name.
;*** ; All files with no extension will be uasc
;
```

Locating the File Type Conversion File

The **pci_cvrt.dat** file that we provide is copied into **/sys/dpci** when the system administrator installs the Domain/PCI Connection server software on Domain.

However, the system administrator may make additional copies of the **pci_cvrt.dat** file in order to avoid conversion problems, one of which we'll discuss in "Storing a PC Application on Domain" later in this chapter.

A customized **pci_cvrt.dat** file contains the extensions and directives that you normally use. Then, if you need to recopy your Domain/PCI Connection server software, the file type conversion file that we supply *will not* overwrite your customized **pci_cvrt.dat** file. It's a good idea to have a customized **pci_cvrt.dat** file in your **user_data** directory under your home directory. Your system administrator can tell you the names of your home directory and your **user_data** directory when your Domain account is created. You might also want to have a customized **pci_cvrt.dat** file in the Domain directory into which you're copying text files.

You can create a **pci_cvrt.dat** file without copying the original file as long as the file you create is called **pci_cvrt.dat**.

Creating Files on Domain from Your PC

When you use your PC to create a file on the Domain system, the Domain/PCI file type conversion file converts the text file to uasc format if the file's extension (or special null extension) is in the **pci_cvrt.dat** file. If the extension is not in **pci_cvrt.dat**, the Domain/PCI Connection creates a file type **hdru** (header undefined), which cannot be read on Domain/OS.

Figure 4-1 shows how a PC text file, **memo.txt**, is converted to a Domain text file when the file is shared. The **pci_cvrt.dat** file contains the extension **txt**, which tells Domain/PCI to convert the file to Domain's uasc file format.

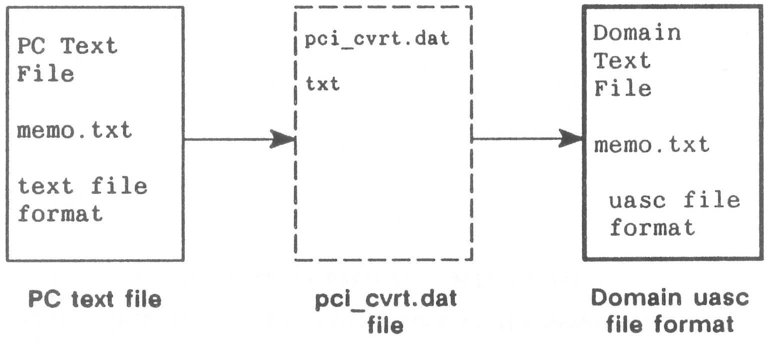


Figure 4-1. Converting a PC Text File to a Domain Text File

Since Domain users cannot read text files that are hrdu file types, if you are creating a text file on your PC that you will share with Domain users, be sure that the file's extension is in the **pci_cvrt.dat** file. This will ensure that the file is converted to a uasc file when it's shared with Domain users.

If the file's extension is not in the **pci_cvrt.dat** file, the file should be either a text file that you do not want to share with the Domain system or the file should be a non-text file.

If a file is not a text file, such as an **.exe** file, and is unintentionally converted to uasc (because its extension is in **pci_cvrt.dat**), some data may be lost when the PC attempts to read the file back. As a result, using the file may have unpredictable results. This may accidentally occur if you attempt to store a PC application on Domain and are not careful about file conversion.

The example in the following section illustrates this important point.

Storing a PC Application on Domain

Let's suppose that you have a word processing application that you would like to store on Domain, but use on your PC. This is a common method of using Domain/PCI, since it enables you to free up space on your PC's own hard disk.

Your word processing application directory contains the following files:

```
wp.exe
dict.txt
readme.doc
```

Your **pci_cvrt.dat** file contains the following entries:

```
txt
doc
```

The **pci_cvrt.dat** file converts all files with a **.txt** or **.doc** extension to uasc. However, suppose that the file **dict.txt** is not a true text file, but is an object file instead, which is used to run the word processor's dictionary. In this case, when you copy the files to Domain, the dictionary file is converted to a text file, and thereby corrupted. When you attempt to use the dictionary on your PC, you may instead see a message similar to: Error in TXT file.

There is a way to fix this problem: create a local **pci_cvrt.dat** file in the directory in which you'll be copying the application. In this local **pci_cvrt.dat** file, eliminate the **.txt** extension from the list. This way, the local **pci_cvrt.dat** file will determine which files are converted and which are not when you copy the application.

The following section shows you where to locate your **pci_cvrt.dat** file to avoid this type of problem.

Which Conversion File Is Used?

The **pci_cvrt.dat** file can be located in any of these locations: the home directory's **user_data** directory, the server's **/sys/dpci** directory, or the local directory into which the text file is being copied. If the **pci_cvrt.dat** file exists in more than one of the following directories, the Domain/PCI Connection looks in each of the following directories for a **pci_cvrt.dat** file, using the following order of precedence:

1. Local directory — read each time a file is written to or read from Domain.
2. **~/user_data** — read during logging in.
3. **/sys/dpci** — read during the startup of the Domain/PCI server.

If the **pci_cvrt.dat** file is not found in any of the above directories, file conversion does not occur.

Converting Files without Extensions

If you want to convert files that do not have extensions, use the **null character** in the **pci_cvrt.dat** file, as shown below:

```
***
```

This converts text files that do not have extensions.

Accessing Files Created on Domain from Your PC

Typically, when you access a Domain file from your PC, if the file is created as a uasc file, it is automatically converted to text file format regardless of its extension.

In certain cases, however, you may find it necessary to prevent automatic file conversion. The directives in the **pci_cvrt.dat** file enable you to control automatic file conversion from uasc to text.

Figure 4-2 shows how a Domain object file, **rndata.bin,** created on a cross compiler as a uasc file, is prevented from be converted to a PC text file when the file is shared. Note that the **pci_cvrt.dat** file contains a directive (**>not bin**) that prevents conversion from occurring. If conversion on a cross compiler file in uasc format were to occur, the file would be corrupted.

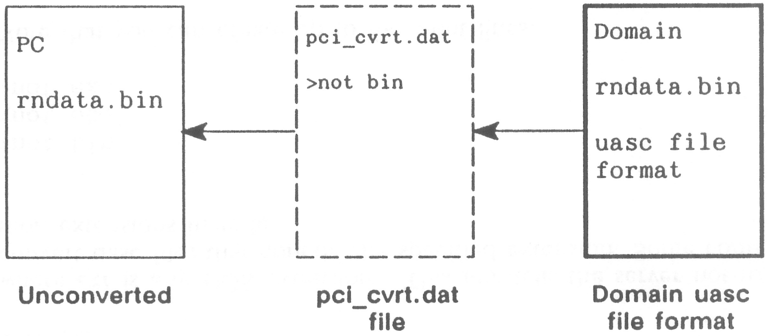


Figure 4-2. Preventing Conversion from a Domain File to a PC

This section describes the cases in which you need to add directives to **pci_cvrt.dat** file to prevent automatic file conversion from occurring.

- Domain® NFS* and Domain®/Access™ files
- Domain uasc files
- Domain cross compiler/assembler files

---

*NFS is a registered trademark of Sun Microsystems Inc.

Domain NFS and Domain/Access Files

The Domain/PCI server allows file sharing through the Domain NFS and Domain/Access products. This means that your PC can access files on the Domain NFS or VAX* VMS* systems.

Domain NFS Files. Domain NFS does not create file types when you access an NFS file on a PC; all Domain NFS files are uasc files. From your PC, you access all files, including binary and executable, in text file format. Normally, the Domain/PCI server automatically converts all uasc files to text files; however, in this case, if the server attempts to convert an NFS binary file, the translation process makes the file unusable.

Therefore, you need to add the following lines to your **pci_cvrt.dat** file if you are reading NFS files to or writing NFS files from a PC:

```
>uid nfs_files       ;NFS file
>uid nfs_dir         ;Directories via NFS
>uid nfs_gate        ;Generic NFS Gateways
```

These directives prevent any PC file with a uid type from being converted when copying to or reading from an NFS machine. Also, PC text files on an NFS machine will contain a carriage return and a linefeed.

If you create your own type manager and use line terminators (other than carriage returns and linefeeds) to represent text files, you will need to use the above directives to prevent the files from being converted.

You can create up to 16 >uid lines.

Domain/Access Files. Domain/Access files use file types; thus, if you are using Domain/Access files, you do not have to add any information to the **pci_cvrt.dat** file to prevent file conversion from occurring. Domain/Access binary files are not converted when you copy a Domain/Access file to your PC.

---

*VAX and VMS are registered trademarks of Digital Equipment Corp.

Domain uasc Files. If you do not want to convert Domain uasc files to a text file, add the following line to the **pci_cvrt.dat** file:

```
>uid uasc
```

This ensures that the file appears on the PC exactly as it is created on Domain.

Domain Cross Compiler/Assembler Files

The Domain cross compilers and assemblers that generate 8086 code create binary and executable files that are uasc files. When you read these files from a PC, or copy the files to a PC, the Domain/PCI server normally converts them to text files. In this case, the conversion process makes the files unusable.

To prevent these uasc files from being converted, use the following directive in the **pci_cvrt.dat** file:

>not *ext*

where *ext* is any DOS extension. This line tells the server not to convert uasc files that contain the specified extension. Some common extensions include:

```
>not bin
>not obj
>not exe
```

Note that you can create up to 64 >not lines.

DOS Backup and Restore Files

When you use the DOS **backup** and **restore** commands on files that are on the Domain system, these commands add control data to the files. Thus, these files *should not* be converted to text files.

To prevent the files from being converted, do the following:

1. Be sure that the connection to the destination drive is the directory above which you want to back up your files. For example, to back up files into a directory below the directory **storeit**, you would connect to **storeit**:

   net connect m: \\wayside\storeit

   When the **backup** command completes, it will create a directory called **\backup** into which it will backup your files. In our example, the full pathname will be

   \\wayside\storeit\backup

2. Create an empty **pci_cvrt.dat** file (on Domain) in the destination directory in which you are backing up the files.

3. Specify the **/a** option to the DOS **backup** command to avoid erasing the **pci_cvrt.dat** file (as well as any other file in the directory). For example:

   ```
   C:>  backup e: m: /a
   ```

   Your files from a directory connected to E: are backed up to M: in the directory created called **\backup.**

4. Before you restore any files, delete the file **pci_cvrt.dat** from the backup directory.

Adding Comment Lines to the File Type Conversion File

You can add comments to your **pci_cvrt.dat** file with a semicolon (;) character. The format of the comment line is

; *comment*

A semicolon or a blank space indicates this is a **comment line**. For instance, the following lines are both comment lines:

```
; This is a comment
   >not obj
```

The following example shows how comments can be used. Note that this file does not use directives and does not convert **.bat** extension files:

```
; Extensions that are converted
asm
txt
;bat
***          ;null extension

.

.

.; nfs type uids --  Are not converted.

;>uid nfs_dir            ; directories via nfs
;>uid nfs_gate           ; generic nfs gateways
;>uid nfs_file           ; nfs files don't get converted
```

Accessing Files

To manipulate Domain files, simply make a connection to a Domain drive, access the directory that contains the file you want, and use a text editor, such as **edlin**, or a word processing application to manipulate the file.

Using a Text Editor

For example, suppose you want to edit a file called **memo.txt** in a Domain directory called **//accounts/payable.** To get to the **memo** file, you could make the following connection:

```
C:>  net connect E: \\accounts\payable
```

Then, switch to drive E and list the directory to ensure that the **memo** file is there:

```
E:> dir
```

You could use the DOS **type** command to display the contents of the file before you edit it, or you could use a text editor, such as **edlin** to edit the file.

```
E:> edlin memo.txt
```

Using a Word Processing Application

Suppose you have a word processing application in a Domain directory called **//wayside/apps**. Also, suppose you keep your documents in another directory called **//node5/docs**. Through your Domain/PCI Connection, you can connect to the directory where the application is, **//wayside/apps**

```
C:> net connect E: \\wayside\apps
```

You can make a second connection to the directory where the documents are, **//node5/docs**

```
C:> net connect F: \\node5\docs
```

Then you can change back to drive E and invoke the word processor. From the word processor, you can specify drive F and enter the name of the document you want to edit. In this case, we want to edit a document called **memo**. For example, in the word processing application, you might type:

```
Drive:  F:              Pathname: memo
```

Depending on your word processing application, the method by which you enter the document's pathname may differ.

Copying Files

The Domain/PCI Connection lets you copy files from your PC to Domain directories and from Domain directories to your PC. When the file you are copying is a text file, the Domain/PCI Connection converts the file to the format appropriate for the system that will use the file.

You can use the DOS **copy** command to copy a file from a Domain directory to a diskette in one of your PC's drives or to your PC's fixed disk.

For example, let's say drive E is connected to the Domain directory **accounts**. To copy the file **accounts\payable\memo** to a diskette in drive A on your PC, type this command:

```
copy  e:\accounts\payable\memo  a:
```

When your PC finishes copying the file, it displays this message:

```
1 File(s) copied
```

You can also use the **copy** command to copy files from a PC diskette or fixed disk to a Domain directory. This command copies the file **status.doc** into the Domain directory **\accounts\reports** (since drive E is connected to **accounts**):

```
copy  a:status.doc  e:reports\status.doc
```

When your PC finishes copying the file, it displays this message:

```
1 File(s) copied
```

Printing Files

This section describes how to print files on a Domain printer from your PC. Before you can begin printing, however, be sure that your system administrator has properly set up the offers file on the Domain system. The system administrator should read Chapter 9 for information about setting up printer offers. You should read this section to learn how the offers file works and what methods you can use to print files.

How the Offers File Works

The **offers file** provides instructions for Domain printers that enable you to associate a *printerdevice* name, such as lpt2 with the *offer_name* your system administrator assigned to the printer in the offers file. This association is made when you execute the **net connect** command using the following format:

net connect *printerdevice*: *offer_name*

where *printerdevice*: is the name of the printer device, such as lpt1, lpt2, or lpt3, and *offer_name* is the name of the printer offer listed in the offers file.

The following examples illustrate how your **net connect** input (from a batch file, such as the client startup file, or from the DOS prompt) and your offers file, **pci_cmd.dat** interact when you want to print a text or graphics file.

Of the following examples, the first file is a text file; the second file is a graphics file that contains IBM Graphics printer commands and is prepared by a spreadsheet application.

Table 4-1. How the Offers File and User Input Interact

	offers file *(from server)*	**net connect** command *(from PC or .bat file)*	**user input** *(from PC)*
1	offer print= -pr p \char	net connect lpt2: print	copy textfile lpt2
2	offer laser= -pr apple	net connect lpt1: laser	copy grafile lpt1

1 Printing a text file

To print the text file, **textfile**, the offers file includes instructions for **print**, the offer name (linked to the **/sys/print** directory of the server node), which sends the output to the default printer (**-pr p**). In the **net connect** command, which you can execute from your client startup file or by typing it in from a DOS prompt, you enter a *printerdevice* **lpt2** and an *offer_name* **print**. Then, when you type

copy textfile lpt2

the **net connect** command associates **lpt2** with **print**. Your offers file provides the instructions for printing the file from the offer **print**.

2 Printing a graphics file

To print the graphics file, **grafile**, the offers file includes instructions for **laser**, the offer name, that send the output to a printer called **apple**. In the **net connect** command, which you can execute from your client startup file or by typing it in from a DOS prompt, you enter a *printerdevice* **lpt1** and an *offer_name* **laser**. Then, when you type

copy grafile lpt1

the **net connect** command associates **lpt1** with **laser**. Your offers file provides the instructions for printing the file from the offer laser.

You can use either the DOS **print** command or the **copy** command to print files on Domain printers. (For more information about these commands, see your DOS Reference Manual.)

Print Command Method

Format

print [*device*:]*pathname*

Arguments

pathname	The directory and filename in which the file you want to print is located.

Options

device:	The drive letter of the directory that contains the file you want to print.

Comments

The first time you use the **print** command after you start your PC, your PC displays this prompt:

```
Name of list device [PRN]:
```

Enter **lpt2** and press <RET>. This installs lpt2 as the default remote printer. When you start the Domain/PCI Connection, the printer device lpt2 is connected to the default Domain printer, specified with the **/sys/print** directory on the server node. If lpt2 was installed previously (in your client startup file), the following message displays:

```
Resident part of PRINT installed
```

When you print subsequent files, your PC sends the files directly to the printer connected to lpt2 without displaying this prompt.

When the file is sent to the printer, a message, such as the following, displays:

```
E:pathname is currently being printed
```

You can print files on a variety of Domain printers. System administrators should read the "Setting Up the Offers File" section in Chapter 9, which describes how to make different types of printers available for printing. You should look at Chapter 5 for the commands that describe how to connect to those additional printers.

Copy Command Method

Format

copy [*device*:]*pathname* *printerdevice*

Arguments

pathname	The directory and filename in which the file you want to print is located.
printerdevice	The name of the printer device, such as **lpt1** **[prn]**, **lpt2**, or **lpt3**.

Options

device: The drive letter of the directory that contains the file you want to print.

Comments

When the file is sent to the printer, a message, such as the following, displays:

```
1 File(s) copied
```

You can print files on a variety of Domain printers. System administrators should read the section "Setting Up the Offers File" in Chapter 9, which describes how to make different types of printers available for printing. You should look at Chapter 5 for the commands that describe how to connect to those additional printers.

Print and Copy Method Examples

Suppose you've made the following connection to the Domain default printer:

```
net connect lpt2: print
```

With drive E connected to the Domain directory **//accounts**, you would print the Domain file **//accounts/loans**, by typing either one of these commands:

```
print  e:loans
```

or

```
copy e:loans lpt2
```

Here's another example. Suppose you've made the following connection to a laser printer on the Domain/OS system:

```
net connect lpt3: mtklsr
```

Your system administrator has created an offer in the offers file for the marketing department's laser printer and named the printer MTKLSR. To print your third quarter budget (also on drive E:) on the marketing printer, you would type:

```
print mtklsr e:3qbudget
```

or

```
copy e:3qbudget lpt3
```

Before you try to execute a **net connect** command to a printer, be sure that the offers file has been set up. You can verify that the offers file has been set up and list offer names of and instructions for the available printers by typing the following command at the DOS prompt:

```
C:>   net offers
```

To verify the printer connections to Domain, type the following at the DOS prompt:

```
C:>   net connect
```

Using Domain Printers

As stated earlier, before you can use a Domain printer, the system administrator must create an offer for it in the offers file and you must use a **net connect** command on your PC to connect to it. System administrators should read the section "Setting Up the Offers File" in Chapter 9, which describes how to make printers available. You should look at Chapter 5 for the commands that describe how to connect PCs to the printers.

This section explains how you specify which Domain printer should be used to print a file.

Using the /sys/print Default Printer

The default Domain system printer, which is linked to the **/sys/print** directory on the server node, is always connected to **lpt2** through the client startup file. For example, the command in the client startup file that connects your PC to the default Domain printer is

```
net connect lpt2: print
```

Using Printers Other than lpt2

To use a printer other than the one connected to lpt2, use the **print** command as follows:

print /d:*printerdevice filename*

where *printerdevice* is the name of the printer devices (lpt1, prn, etc.), and *filename* is the name of the file you want to print. For example, the following command uses the printer named lpt1 in the offers file to print the file **reports\status.doc**:

print /d:lpt1 e:reports\status.doc

Printing from Application Programs

Most application programs use a default PC printer device for printing. When you print directly from the application, your PC sends the data to the printer assigned to that device. If the device is connected to a Domain printer, the data is printed on the Domain printer.

For example, if your application uses the printer device prn for printing from within the application, and prn is connected to a Domain printer, when you print from the application, the data is printed on the Domain printer.

Changing the Application's Default Printer

If the application's default printer device is connected to a Domain printer, and you want to use a local printer instead, disconnect the printer device from the Domain printer with the **net disconnect** command.

Printing Graphics from an Application

If you want to print graphics from an application, the application may require that you specify the kind of printer you will be using for printing. Because the Domain/PCI Connection emulates the IBM PC graphics printer, you should specify the **IBM PC graphics printer** as the printer type.

Saving Data to Print Later

Some applications let you save data into a file that you can print at a later time. When you want to create such a file and save it on the Domain system, make sure that the filename's extension is listed in the file type conversion file, **pci_cvrt.dat**, so that the file will be correctly saved in uasc format.

How to List Files in a Print Queue If They Don't Display

Some applications send print files to the printer but you cannot see the files in the print queue until you exit from the application. This occurs because these print files are missing the end-of-file (EOF) indicator. To add an EOF character to the file, send the file to the printer and then press <CTRL><ALT><PRTSC>. The file is then immediately queued to the printer.

NOTE: Do not press <CTRL><ALT><PRTSC> *before* you send the file because it separates the print file into several files.

What's Next?

In this chapter, you learned how to handle file conversion whe sharing files with Domain. You also learned how to access files wit a DOS text editor or from applications, how to copy files to an from Domain, and how to print PC files on Domain printers.

The next chapter describes all the net commands that enable you t use the Domain/PCI Connection.

* * *

Chapter 5

The Domain/PCI net Command Reference

The Domain/PCI Connection lets you access the resources of the Domain system, such as directories, files, and printers. The resources of the Domain system are known as **remote resources**. This chapter describes the commands that enable you to access these resources.

For each command, we give its format, a description, and examples. This guide provides two ways to look up a command: by task or by command name. We first summarize the command by task, in Table 5-1, then we present each command alphabetically and provide a full description.

net Commands by Task

Table 5–1 lists the net commands by task. A detailed summary of each net command follows this table.

Table 5–1. The net Commands

How do I ...?	net Commands	Page
Connect drives to Domain directories	net connect*	5–5
Connect to a Domain printer	net connect*	5–9
Disconnect from a Domain directory	net disconnect*	5–15
Disconnect from a Domain printer	net disconnect*	5–17
Get help	net help	5–18
List PC connections	net connect*	5–11
List the contents of offers file	net offers	5–24
Load the network software	net start	5–28
Log in	net login	5–20
Log out	net logout	5–22
Resume connections to Domain resources	net continue	5–13
Suspend connections to Domain resources	net pause	5–26

*You can also use the **net use** command for versions 1.0 and 2.0 of Domain/PCI SIO connection products.

net Commands by Name

This section describes all the net commands in detail. To find a command, look up the name of the command alphabetically.

net connect — Connect to a Domain directory

Purpose

Enables you to connect your PC to a Domain directory.

Comments

The **net connect** command allows you to connect your PC to a Domain directory through a virtual drive. You can connect up to ten different directory connections.

You must run the **net start** and **net login** commands before you can use the initial **net connect** command. You may want to put a few customized **net connection** commands in your client startup file so that it makes connections to the network automatically. Refer to Chapter 2 and Appendix D for more information.

Format

net connect *device*: [*pathname*] [[**-on** *server_id*] [**-login** *user_name*]]

Arguments

device:

The name of the virtual drive you want to connect to the Domain directory. For example, D:.

If you attempt to connect a drive outside the range you specified in the **lastdrive** command (described in Chapter 3), your PC displays this message:

```
Invalid device
```

Options

pathname

The connection point in the Domain file system. The directory specified in *pathname* becomes the root of the network drive specified in *device*:.

The format of *pathname* is similar to Domain/OS pathnames. However, only backslashes (\\) are legal as directory separators. For example: **\\node5\david**. Also, note that directory names in *pathname* are not restricted to the DOS 8.3 characters format (eight characters followed by a period and a three-character extension) but can be any Domain/OS name that is all lowercase. (Mixed case and uppercase names aren't supported by DOS.)

If you do not use the *pathname* option, the connection point defaults to your **home directory**. The home directory is your personal directory that your system administrator sets up for you when you get a user name and password.

Your PC's access to the Domain file system is limited to the connected directory and to files and directories below that directory.

-on *server_id*

Enables you to override the *server_id* that you specified in the **net login** command. The *server_id* is the hexadecimal ID of the node. For example, **-on 18dfc**.

-login *user_name*

Enables you to override the user identity that you specified in the **net login** command.

Example

Suppose you have already started the Domain/PCI-Ring Connection and you have logged into the server on node 12345 with the following commands:

```
net start dpciring
net login beth -on 12345
```

Once the product identification information displays, and after you have entered your password, you log in as **beth**. Now you want to connect drive E of your PC to a Domain directory called **//acts/payable** in the Domain file system. You execute the command:

```
net connect e: \\acts\payable
```

When the connection is completed, your PC displays this message:

```
Command completed successfully.
```

Your user name (**beth**) and the server ID (**12345**) for drive E are inherited from your earlier net login.

You can now use the files in the **\\acts\payable** directory and any subdirectory under **\\acts\payable** the same way you use files on a diskette in one of your PC's drives or files on your PC's fixed disk.

However, if you want to override the default user account you set when you logged in, you can use the **-login** option. For example, suppose you want to change your permission rights to system administration permission rights and connect to the **//acts/payable** directory from drive F. In this case, the **net connect** command looks like this:

```
net connect f: \\acts\payable -on 12345 -login
Please log in:  beth.sys_admin
Password: . . . . .
```

When the connection is completed, your PC displays this message:

```
Command completed successfully.
```

followed by a DOS prompt.

You can now access files from drive E as beth and you can access files from drive F as beth.sys_admin.

net connect — Connect to a Domain printer

Purpose

Enables you to connect your PC to a Domain printer.

Comments

You must run the **net start** and **net login** commands before you can use the **net connect** commands. The **net connect** command associates the name of a PC printer device with the Domain printer.

When you start up the Domain/PCI Connection, the default client startup file connects lpt2 to the default Domain system printer; that is, the one specified in the **/sys/print** directory. However, you can choose other printer devices and other printer offers for your remote printer connections.

Format

net connect *printerdevice*: *offer_name* [**-on** *server_id*]

Arguments

printerdevice	The name of a printer device; the name may be one of the following: **lpt1 [prn]** **lpt2** **lpt3** You can connect a printer device to only one printer at a time. (For more information on device names, see your DOS Reference Manual.)
offer_name	The name of the Domain printer offer to which you want to connect.

The names of the Domain printers that you can connect to are established in the offers file, **pci_cmd.dat**, on the server node. To list the contents of the offers file, type **net offers** at the DOS prompt. Also, for more information refer to the **net offers** command in this chapter.

Options

-on *server_id*	Enables you to override the *server_id* that you specified in the **net login** command. The syntax of *server_id* is the hexadecimal ID of the node. For example, **-on 18dfc**. If you do not know the ID of the printer server to which you want to connect, ask your system administrator.

Example

The command below connects the printer named **lpt3** to the laser printer with the offer name called **laser**:

```
net connect lpt3: laser -on 18dfc
```

The name and print options for this printer, **laser**, is established in the offers file.

When the connection is complete, your PC displays this message:

```
Command completed successfully.
```

net connect — List PC connections

Purpose

Enables you to list the names of the Domain resources connected to your PC's devices.

Comments

When you use the **net connect** command with no options, your PC displays a list of connected devices and the current path to which you are connected.

Format

```
net connect
```

Example

```
C:> net connect

Connect - DPCI Connection Utility Version x.xx

Default node id;  18DFC
Default username:  BETH

Local Name             Remote Name

D:                     //18DFC/BETH
Path is:               //

E:                     //18DFC/BETH
Path is:               \\ACTS\BETH\PAYABLE

F:                     //18DFC/BETH.SYS_ADMIN
Path is:               \\ACTS\BETH\PAYABLE

G:                     //18DFC/WAYSIDE

LPT2:                  //18DFC/PRINT

Command completed successfully.
```

In this example, we see that drive D is connected to the top-level Domain directory. We can also see that drives E and F are connected to the same Domain/OS directory, **//acts/beth/payable**. However, on drive E, Beth is logged on as a user, and on drive F, Beth is logged on as a system administrator. Drive G is connected to another directory. The default Domain printer **lpt2:** is connected to the server to which Beth initially logged on.

net continue — Resume connections to Domain resources

Purpose

Enables you to restore your PC's connections to Domain resources after you have suspended the connections with the **net pause** command (described in this chapter).

Format

net cont[inue]{ drdr | prdr }

or

net cont[inue]{ disk redirection | print redirection }

Arguments

drdr | disk redirection
Resumes the connections between your PC's drive devices and Domain directories.

prdr | print redirection
Resumes the connections between your PC's printer devices and Domain printers.

Example

Let's say you have typed the command **net pause drdr** to suspend the connections between your PC's drive devices and Domain directories. From a DOS prompt, such as C:, type the following command to restore the connections:

```
C:> net cont drdr
```

Let's say you have typed the command **net pause prdr** to suspend the connections between your PC's printer devices and Domain printers. Type the following command to restore the connections:

```
C:> net cont prdr
```

net disconnect — Disconnect from a Domain directory

Purpose

Enables you to disconnect your PC to a Domain directory.

Comments

You may want to disconnect a PC drive device from a Domain directory so that you can connect the drive to another Domain directory or use the drive locally.

You should change to another drive *before* you try to disconnect from the current drive. If you try to disconnect from the drive you're in, your PC displays this message:

```
Current drive is no longer valid>
```

If you see this message, enter a valid drive at the prompt and press <RET>.

Format

net disconnect *device*:

Arguments

device: The name of the drive you want to disconnect.

Example

To disconnect drive E of your PC from its associated Domain directory, first make sure that drive E is not your current drive. Then type the following command to disconnect drive E:

```
C:> net disconnect e:
```

You can now connect drive E to another Domain directory.

> **NOTE:** You can use the **net connect** *device*: **-d** command in place of **net disconnect**.

---

net disconnect — Disconnect from a Domain printer

Purpose

Enables you to disconnect your PC printer from a Domain printer.

Format

net disconnect *printerdevice*:

Arguments

printerdevice: The name of the printer you want to disconnect.

Example

The following command disconnects the PC printer named lpt1 from its associated Domain printer:

```
net disconnect lpt1:
```

When the printer is disconnected, your PC displays this message:

```
Command completed successfully.
```

> **NOTE:** You can use the **net connect** *device*: **-d** command in place of **net disconnect.**

net help — Display information about commands

Purpose

Enables you to obtain information about how to use the net commands.

Format

When you type just the **net help** command, your PC displays information about the Domain/PCI net commands:

```
net help
```

To get information about a specific command, type

```
net command help
```

where *command* is the name of a net command.

Arguments

The valid *command* names are: **use**, **connect**, **login**, **pause**, **continue**.

Example

When you type the following command, your PC displays information about the **net pause** command:

```
net pause help
```

NOTE: The Domain/PCI Connection help files are contained in the **\dpci** directory. You can use the DOS **append** command to set a search path to the **\dpci** directory. The **append** command is described in Chapter 4.

net login — Log in

Purpose

Enables you to log in.

Comments

You must run the **net start** command before you can use the **net login** command. The **net start** command is described in this chapter.

Before connecting your PC to any Domain directory, you must first use the **net login** command. Your system administrator will set up an account for you (with a username and a password) so that you can log in. If you are already using Domain, you use the same user ID and password when you're logging in on your PC.

The **net login** command sets the default user account for your Domain/PCI Connection and also identifies your default server on a Domain/PCI network connection product. Each connection you make to a server has the default user account associated with your username and password. You can override the user account by using the **net connect** command with the **–login** option. The **net connect** command is described in this chapter.

Format

net login [*user_id*[. *project*[. *org*]]] [**–on** *server_id*]

Options

user_id	Specify your user ID. If you omit this option, you are prompted to log in interactively.
project	Specify your project ID (if you have one). This ID may be separated from *user_id* by a space or period.
org	Specify your organization ID (if you have one). This ID may be separated from *user_id* by a space or period.
–on *server_id*	This option enables you to specify the hexadecimal node ID of the machine on which your Domain/PCI server is running. The syntax of *server_id* is the hexadecimal node ID, for example, **–on 18dfc**. If you do not know the node ID of the server to which you want to connect, ask your system administrator.

NOTE: When you log in, you may need to wait about 30 seconds before the login command successfully completes. Do not execute the **net login** command again while waiting for the login to be completed.

Examples

```
net login  beth.sys_admin.sales -on 18dfc
Password:  . . . . .
```

In this example, the user, Beth, includes the full username option. Note that the user must provide a password.

```
net login -on 18dfc
Please log in:  beth
Password:  . . . . .
```

In this example, the user omits the *user_id* option and is prompted to log in.

net logout — Log out

Purpose

Enables you to log out of Domain/PCI, break all connections, and remove the server processes created when you logged in. Logging out ensures that no other individual will be able to access the files and directories protected under your user account. This is an added security feature that prevents others from accessing your files. Also, this command conserves server resources by eliminating the server processes that were created on logging in.

Comments

Whenever you are not using the Domain/PCI Connection and planning to leave your PC unattended for a period of time, you should log out of Domain/PCI with this command.

Change to a local DOS drive *before* you try to log out. If you try to log out from a virtual drive, your PC displays these messages:

```
Session closed

Current drive is no longer valid>
```

If you see these messages, enter a valid local drive, such as C:, and press <RET>.

When you log out, **net logout** breaks all connections to Domain resources and removes the server processes created when you logged in. To reconnect, you must start the Domain/PCI Connection and log in again.

Format

net logout

Example

The following example illustrates how the **net logout** command breaks all connections and removes the server process.

Before logging out, type the **net connect** command, which lists your Domain/PCI connections. In DTERM, you can check the server process by typing a **pst** command, which lists current processes. Note that the server process ID is associated with the PC's node ID:

```
.
.
.
dpcinet_server_ed6A
.
.
.
```

Log out of DTERM. In the DOS environment, type the **net logout** command.

```
C:> net logout
```

You then type the **net connect** command again:

```
C:> net connect
```

You will see that there are no longer any connections.

net offers — List the offers

Purpose

Enables you to list the contents of the offers file, **/sys/dpci/pci_cmd.dat** on the server node.

Format

net offers

Example

To find out the offer names of the printers, type the **net offers** command at the DOS prompt

```
C:> net offers
```

The offers file displays on your PC

```
OFFER PRINT=-PRP \CHAR
OFFER LASERW=-PRLASERAPP -PCPRINT -FONT Helvetica
-TRANSPARENT \CHAR
Command completed successfully.
```

In this example, **print** is the offer name of the default Domain printer and **laserw** is the offer name of the laser printer. The following options may help you to determine the type of printer associated with the offer:

-PR*printer_name*	Specifies the name of the printer. A **P** as *printer_name* indicates that this is the default printer, which is connected to the **/sys/print** directory.
\CHAR	Indicates that this is a printer offer.
-PCPRINT	Indicates that the file is converted to PostScript* before it's printed.
-FONT	Specifies the font to be used for printing when you include the **-pcprint** option.

---

*PostScript is a registered trademark of Adobe Systems Incorporated.

net pause — Suspend connections to Domain resources

Purpose

Enables you to suspend the connections between your virtual drive devices or printer devices and Domain resources. The **net continue** command, described in this chapter, restores your drive or printer connections to the Domain resources.

Comments

Occasionally you may want to suspend the connections between your PC's devices and Domain resources so that you can use the PC devices for connecting to local disks, directories, or printers. For example, you might want to use the PC printer device lpt1 for printing on a printer connected to your PC.

You should change to a local DOS drive *before* you try to disconnect from the current drive. If you try to disconnect from within a virtual drive, your PC displays this message:

```
Current drive is no longer valid>
```

If you see this message, enter a valid local DOS drive, such as C:, and press <RET>.

Format

```
net pause {drdr | prdr}
```

or

```
net pause  {disk redirection | print redirection}
```

Arguments

drdr | disk redirection
Suspends all connections between your PC's drive devices and Domain directories.

prdr | print redirection
Suspends all connections between your PC's printer devices and Domain printers.

Example

Let's say that the drive device named B is connected to a Domain directory, and you now want to use B for accessing your PC's diskette drive. Type the following command to suspend all connections between PC drive devices and Domain directories so that you can use drive B locally:

```
net pause drdr
```

Suppose you have a printer device named lpt1 connected to a Domain printer, and you now want to connect lpt1 to a local printer. Type the following command to suspend all connections between PC printer devices and Domain printers so that you can connect lpt1 locally:

```
net pause prdr
```

net start — Start the network without connections

Purpose

Enables you to invoke the network without making any specific connections.

Comments

You must run the **net start** command before you can use the **net login** or the **net connect** commands.

Format

net start dpci*product* [*–retries zz*] [*–line n*] [*–baud yyyy*]

where *product* is the product name as listed in Table 1–1.

Options

Domain/PCI Direct Connection Products

–retries *zz* — Specifies the number of times the PC should try to execute a command again after encountering an error. The default number is 60. The maximum number of retries is 250. If you are on a busy network, you may want to use a number greater than the default to ensure that you stay connected to the network.

Domain/PCI SIO Connection Products

–line *n* — Indicates the number of the asynchronous communication adapter connector where the cable for the Domain/PCI Connection server is plugged in; this number *n* can be 1 or 2. **1** indicates COM1 (PRIMARY or LINE 1) and **2** indicates COM2 (ALTERNATE or LINE 2).

–baud *yyyy* — Specifies the baud rate, where *yyyy* is the rate. The value of *yyyy* can be 300, 600, 1200, 2400, or 4800.

You only need to specify this option if you have established a baud rate on the server node that is different from 9600. (System administrators, see Chapter 9.)

–retries *zz* — Specifies the number of times the PC should try to execute a command again after encountering an error. The maximum number of retries is 250. The default number is 10. If you are on a busy network, you may want to use a number greater than the default to ensure that you stay connected to the network.

Example

The following example starts Domain/PCI–1 without making any connections:

```
net start dpci1 -retries 25 -line 2
```

net use — Connect to a Domain directory

Purpose

Enables you to connect your PC to Domain drives. This command is only for compatibility with earlier versions.

Comments

Your system administrator must first define the drives or resources before you can connect to them. The resource must be defined in the offers file, **/sys/dpci/pci_cmd.dat**, which is located on the server node.

Once a Domain resource is defined in the **pci_cmd.dat** offers file, and you have started the Domain/PCI Connection, you can use the **net use** command to connect your PC to the resource. After you have connected to a resource, you can use that resource.

You connect your PC to a Domain resource by associating one of your PC's printer or drive devices with that resource. You can connect a PC device to only one Domain resource at a time, but you can connect different PC devices to the same Domain resource.

The **net use** command associates the name of a drive with a Domain directory (offer) listed in the offers file so that you can access that directory from your PC. Your PC's access to the Domain file system is limited to the connected directory and to files and directories under that directory. A **net use** command connection will only give you **user.none.none** rights.

Format

```
net use device: \\domain\offer_name [password]
```

Arguments

device	The name of the virtual drive you want to connect to the Domain directory. You can connect a drive to only one Domain directory at a time. If you attempt to connect a drive not included in your **lastdrive** command of your **\config**.sys file, your PC displays this message: `Invalid device`
domain	The name of the server to which you want to connect. With the Domain/PCI SIO connection products, this name is always **domain**.
offer_name	The name of the offer to which you want to connect. The offer names for the Domain directories that you can connect to are established in the offers file, **pci_cmd.dat**.

Options

password	The system administrator can use a password to limit access to directories. This password is not associated with your Domain/OS registry password.

Example

Let's say you want to connect drive E of your PC to a Domain directory specified as **accounts** in the offers file. In the **net use** command, the *device* is E and *offer_name* is **accounts**. Therefore, to perform the connection, make sure you have started the Domain/PCI SIO connection product.

Type the following at the DOS prompt:

```
net use e: \\domain\accounts
```

When the connection is completed, your PC displays this message:

```
Command completed successfully.
```

You can now use the files in the **accounts** directory the same way you use files on a diskette in one of your PC's drives or files on your PC's fixed disk.

net use — Connect to a Domain printer

Purpose

Enables you to associate the name of a PC printer device with the Domain printer. This command is only for compatibility with earlier versions.

Comments

When you start up the Domain/PCI Connection, the default startup file connects lpt2 to the default Domain system printer; that is, the one specified with the **/sys/print** directory on the server node. However, you can choose other printer devices and other printer offers for your remote printer connections.

Format

net use *printerdevice*: **domain***printername*

Arguments

printerdevice	The name of a printer device; the name may be one of the following: **lpt1 [prn]** **lpt2** **lpt3** You can connect a printer device to only one printer at a time. (For more information on device names, see your DOS Reference Manual.)
domain	The name of the server to which you want to connect. The name is always **domain**.

printername The name of the Domain printer offer to which you want to connect.

The names of the Domain printers that you can connect to are established in the offers file, **pci_cmd.dat.**

If you do not know the ID of the printer server to which you want to connect, ask your system administrator.

Example

net use lpt1: \\domain\spin

This command connects the PC printer named **lpt1** to the printer with the offers name called **spin**.

The name and print options for this printer, **spin**, is established in the offers file, **pci_cmd.dat.**

When the connection is complete, your PC displays this message:

```
Command completed successfully.
```

net use — Disconnect from a Domain directory

Purpose

Enables you to disconnect a PC drive device from a Domain directory so that you can connect the drive to another Domain directory or use the drive locally. This command is only for compatibility with earlier versions.

Comments

You cannot disconnect a drive while it is the current drive. If you try to do so, your PC displays this message:

```
Cannot disconnect from current drive
```

If you see this message, change to another drive and repeat the disconnect command.

Format

net use ***device*****: /d**

where *device* is the name of the drive you want to disconnect.

Example

To disconnect drive E of your PC from its associated Domain directory, first make sure that your current drive is not drive E. Then type the following command to disconnect drive E:

net use e: /d

You can now connect drive E to another Domain directory or use the drive locally.

net use — Disconnect from a Domain printer

Purpose

Enables you to disconnect from a Domain printer. This command is only for compatibility with earlier versions.

Format

net use *device*: **/d**

where *device* is the name of the printer device from which you want to disconnect.

Example

The following command disconnects the PC printer named **lpt1** from its associated Domain printer:

```
net use lpt1: /d
```

When the printer is disconnected, your PC displays this message:

```
Command completed successfully.
```

net use — List PC connections

Purpose

Enables you to see the names of the Domain resources connected to your PC's devices by using the **net use** command with no arguments. This command is only for compatibility with earlier versions.

Comments

When you enter the **net use** command, your PC displays a list of PC device names and their connected Domain resources. The Domain resources are defined in the listing as "Network Names."

Format

net use

Example

```
A:> net use


            Local       Network
Status      Device      Name
------------------------------------------------
            D:          \\DOMAIN\AEGIS
            LPT2        \\DOMAIN\PRINT

Command completed successfully.
```

In this example, we see that drive device **D** is connected to the top-level directory of Domain/OS, and the printer **lpt2** is connected to the default Domain system printer; that is, the one specified with the **/sys/print** directory or link on the server node.

NOTE: If the Domain/PCI SIO connection products cannot access the Domain system resource, "ERROR" 34 is displayed in the "Status" column.

◆

Chapter 6

Getting Started with DTERM

DTERM (Domain Terminal) is the terminal emulation program provided with the Domain/PCI Connection. DTERM lets you use your PC to emulate a VT100 terminal.

When your PC emulates a VT100 terminal, you can run applications designed for the VT100, you can access Domain system services such as DPSS/Mail, or you can use Domain/OS shell commands. Since DTERM does not allow graphics or window capabilities on your PC, you cannot use Display Manager (DM) commands.

With DTERM, you can toggle between VT100 terminal emulation and DOS.

If you are not yet familiar with the Domain system, or if you have questions about the Domain system concepts presented in this chapter, please refer to *Getting Started with Domain/OS* or *Using Your SysV Environment*, *Using Your BSD Environment*, or *Using Your Aegis Environment*.

This chapter explains how to start and stop DTERM, and describes the DTERM display. Chapter 7 describes how to use shell commands and Domain system services with DTERM, and Chapter 8 explains how to use DTERM for running applications designed for VT100 terminals.

Starting DTERM

After you have started the Domain/PCI Connection (see Chapter 2), you can start DTERM by typing the **dterm** command at the DOS prompt.

Format

dterm [*options*]

Options

-alt

Selects the alternate Hot Key feature. The Hot Key feature lets you toggle between a Domain/OS shell and DOS. See "Using the Hot Key Feature to Switch between Domain/OS and DOS," later in this chapter for more information.

-program "*pathname* [*argument...*]"

Executes the program specified by *pathname*. The pathname must be enclosed within double quotes. For example,

dterm -program "/bsd4.3/bin/start_csh"

starts 4.3BSD. See "Starting Up DTERM with a Specific Program" later in this chapter.

-gr

Allows you to run programs that use graphics mode while DTERM is running in the background.

-on *server_id*

Enables you to override the *server_id* that you specified in the **net login** command. The *server_id* is the hexadecimal node ID. For example, **-on 18dfc**. If you use this option, be sure to read the section, "If You Use the -on or -login Option" in this chapter.

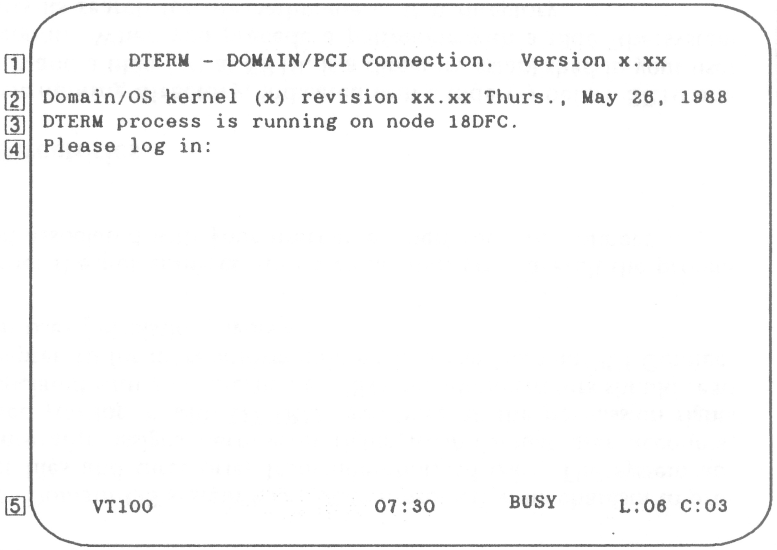
If you do not know the node ID of the server to which you want to connect, ask your system administrator.

-login

Enables you to override the default username and password identity (**SID**) that you initially set with the **net login** command. If you use this option, be sure to read the section, "If You Use the -on or -login Option" in this chapter.

If you omit the **-login** option, the SID for your DTERM process is the one you created with the **net login** command.

After you type the **dterm** command and press <RET>, you see the DTERM display, as shown in Figure 6-1.



```
      DTERM - DOMAIN/PCI Connection.  Version x.xx
Domain/OS kernel (x) revision xx.xx Thurs., May 26, 1988
DTERM process is running on node 18DFC.
Please log in:

  VT100                    07:30      BUSY      L:06 C:03
```

Figure 6-1. DTERM Display before Login

The display includes five lines of information:

1 The DTERM version number.

2 The Domain/OS version number, its date, and time (the time is not displayed in this example). You can also type **/com/bldt** at the prompt to display this information from the server to which you are connected.

3 The ID of the server to which you are connected. You can also type **/com/bldt** at the shell prompt to display the ID of the server to which you are connected.

4 The login prompt appears only if you used the **–on** or **–login** overrides, or if you try to run DTERM without using the **net login** command.

5 The status line. (See "The Status Line," later in this chapter.)

If You Use the –on or –login Options

If you use the **–on** or **–login** option when you invoke DTERM or if you haven't used the **net login** command, the "Please log in:" prompt appears. This prompt indicates that DTERM has made the connection and the Domain system is waiting for you to enter your Domain username and password. If you don't know what to enter, ask your system administrator to define a user account and password for you.

To log in, follow these steps:

1. Type your username next to the prompt and press <RET>:

   `Please log in:` *username*

   Your PC then displays the password prompt:

   `Password:`

2. Enter your password and press <RET> to finish the log–in sequence.

Once you have successfully logged in, the PC displays a Domain/OS prompt. The prompt will be either a $ or a %, depending on the environment of the server to which you are connected. If the server contains a **/com/sh** directory, the prompt will be an Aegis prompt ($). If the server does not contain a **/com/sh** directory, the prompt will be a $ (from **/bin/sh**) or % (from **/bin/csh**).

Account Protection

The Domain/OS system has a system protection mechanism to protect files and directories from unauthorized use. The system administrator assigns permission rights to individual user accounts. Once you log in with DTERM, you have all the permission rights associated with your username. (System administrators should read Chapter 10 for more information on how the Domain/PCI Connection uses permission rights.)

Under the **net connect –login** command, you have all the protection associated with your username when you first connect.

Naming Directories

Your **naming directory**, indicated with a tilde (~) before SR10 or a slash and a tilde (~/) at SR10, is a directory established in your user account. When you precede a pathname with a tilde, the system starts its search for the pathname at that directory.

See *Getting Started with Domain/OS* for more information about naming directories.

Startup Scripts

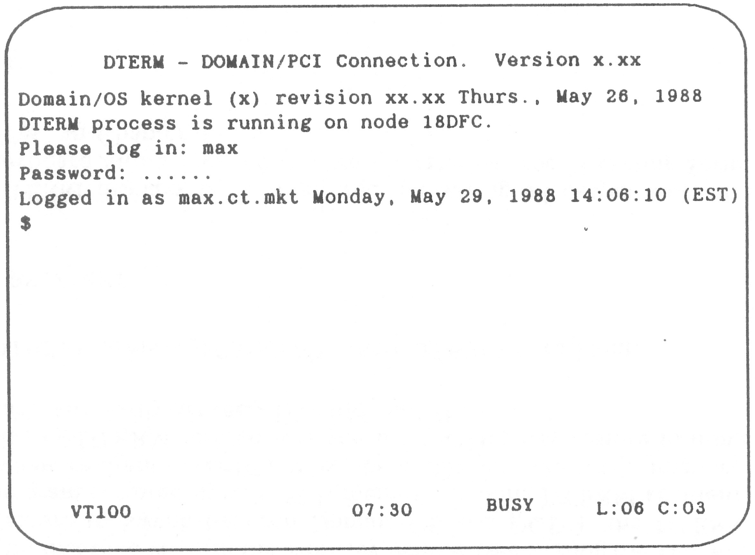
A **startup script** is a file that you create in the **user_data** subdirectory of your log–in home directory on the Domain system. The file contains Display Manager (DM) commands that you want executed when you log in to the Domain system.

When you log in to the Domain system with DTERM, the DM tries to execute the commands in your startup script, if the file exists. Your startup script might contain commands for creating background processes or for establishing shell variables.

Because DTERM does not provide windowing and graphics capabilities, the DM does not execute commands in your startup script that require these capabilities. See Chapter 7 for more information on using scripts.

The DTERM Display

Figure 6–2 shows an example of what you see after you log in with DTERM.



```
        DTERM - DOMAIN/PCI Connection.  Version x.xx
Domain/OS kernel (x) revision xx.xx Thurs., May 26, 1988
DTERM process is running on node 18DFC.
Please log in: max
Password: ......
Logged in as max.ct.mkt Monday, May 29, 1988 14:06:10 (EST)
$









   VT100                       07:30       BUSY      L:06 C:03
```

Figure 6–2. DTERM Display after Login

DTERM supports a screen size of 24 lines by 80 characters.

The Status Line

The **status line** is the highlighted line at the bottom of your PC's screen. The status line displays messages and provides information about the DTERM environment. That information is described here, along with an illustration of where on the status line the information appears.

DTERM continuously displays the time of day using 24–hour format; that is, 7:30 a.m. is displayed as 7:30, and 7:30 p.m. is displayed as 19:30. The colon in the time display blinks on and off at one–second intervals.

7:30

DTERM displays the current line (L) and column (C) position of the cursor. The upper leftmost position is line 1, column 1, which is displayed this way:

L:01 C:01

While you are typing input at the keyboard, the status line displays **BUSY**. Most of the time **BUSY** flashes on and off as you type; however, if too many keystrokes are entered, **BUSY** remains on until all input keystrokes are displayed. During the time **BUSY** is on, you cannot enter other input.

BUSY

When you press <NUMLOCK> once, keys 0 through 9 on the numeric keypad are in the numeric state. This means that the keys are interpreted as numbers rather than cursor–control keys. When you press <NUMLOCK> again, the keys are interpreted as cursor–control keys. When the keys are in the numeric state, DTERM displays **NUM**.

NUM

While you are in DTERM, pressing <CTRL> S stops information from scrolling on the screen, and pressing <CTRL> S again restarts the scrolling. When you have pressed <CTRL> S to stop information from scrolling, DTERM displays **XOFF**.

```
XOFF
```

The first time you use the Hot Key feature (see "Using the Hot Key Feature to Switch between Domain/OS and DOS"), the DTERM program is stored in your PC's memory, making DTERM **resident**. When resident, DTERM uses about 30K of your PC's memory. Having DTERM resident lets you use the Hot Key feature to move back and forth between DTERM and DOS.

DTERM displays **RESIDENT** when DTERM is resident.

```
RESIDENT
```

DTERM displays **OUT** when you are saving all output from the keyboard into a file. (For more information, see "Creating Script Files" in Chapter 7.)

```
OUT
```

DTERM displays **IN** when you are using the contents of a file for input instead of the keyboard. (For more information, see "Creating Script Files" in Chapter 7.)

```
IN
```

DTERM displays **VT100** on the status line as displayed here.

```
VT100
```

DTERM displays **WRAP** when you have enabled automatic line wrap. (You control line wrap in the SET–UP B menu, as described in the "Setting Screen Characteristics" section in Chapter 8.)

```
WRAP
```

Using the Hot Key Feature to Switch between Domain/OS and DOS

The Hot Key feature of DTERM lets you toggle between a Domain/OS shell and DOS (or an application program running under DOS) without losing your place on either side. Depending on your needs, you can use either the Hot Key feature or the Alternate Hot Key feature. If you're using another PC communication product in addition to the Domain/PCI Connection, the Alternate Hot Key feature prevents the possibility of having the same Hot Key sequence for both products. Both methods are described in this section.

NOTE: Do not use Hot Key from the VT100 set–up modes, described in Chapter 8.

How to Use the Hot Key

To use Hot Key, press <LEFT SHIFT><RIGHT SHIFT> (both keys at the same time).

You can use Hot Key as often as you like. DTERM always saves your place in DOS or Domain/OS and redraws the screen at the point you left. Because DTERM maintains a screen buffer to redraw the screen, it holds your place in DOS and does not execute any further instructions until you switch back to DOS using the Hot Key feature.

How to Use the Alternate Hot Key

DTERM also provides an alternate Hot Key sequence, <ALT><RIGHT SHIFT>. If you are using another PC communication product in addition to the Domain/PCI Connection, this prevents the possibility of having the same Hot Key sequence for both products. Select the alternate Hot Key sequence when you first enter the DTERM program by typing the following at the DOS prompt:

```
dterm -alt
```

Then, to toggle out of DTERM, press the <ALT><RIGHT SHIFT> keys.

For more information about starting the DTERM program, see the section "Starting DTERM" in this chapter.

Stopping DTERM

To stop DTERM, you can use one of two methods: the **logout** command method or the <SHIFT><ESC> method.

The logout Command Method

To use the **logout** command method, type the **logout** command at the DTERM prompt:

```
$ logout
```

When you do so, DTERM first displays a message indicating that the DTERM program is stopped, and then displays the DOS prompt:

```
?(sh) process stop (OS/fault handler)

>
```

The <SHIFT><ESC> Method

Alternatively, you can stop DTERM by pressing <SHIFT><ESC>. When you do so, DTERM displays this prompt:

```
Do you wish to terminate the emulator (y or n)?
```

Answer **Y** (or y) to exit from DTERM and return to DOS. Answer **N** (or **n**) to cancel the exit request.

You can restart DTERM at any time, as described earlier in this chapter.

NOTE: If you've used the Hot Key feature to move back and forth between DTERM and DOS, make sure that DOS is at the > prompt before you stop DTERM. If DOS is not at the > prompt, the PC may crash when you try to stop DTERM. (When your PC crashes you must reboot your PC.)

If you've used the Hot Key feature (see "Using the Hot Key Feature" earlier in this chapter), stopping DTERM regains approximately 30K of memory for your PC.

Stopping DTERM also stops the special process created on the server node for running DTERM.

Starting Up DTERM with a Specific Program

The **-program** option of the **dterm** command lets you specify a program for DTERM to execute immediately after you log in to DTERM.

Format

```
dterm -program "pathname [argument...]"
```

Arguments

pathname	Specifies the full pathname of the program to b executed.

Options

argument	Specifies any arguments or options that you wa passed to the program.

Note that you must enclose *pathname* and *argument* in doubl quotes.

When you use the **-program** option, DTERM starts up and exe cutes the program specified by *pathname*, passing to the progra any arguments you specified.

Example

Let's say you want to use DTERM with a BSD shell from a serve that normally defaults to Aegis. First, start DTERM with the **-pro gram** option, specifying the program that starts a BSD shell. Fro the DOS prompt, type the following:

```
dterm -program "/bsd4.3/bin/start_csh"
```

DTERM executes **/bsd4.3/bin/start_csh**, which starts a BSD 4. shell. You then see the prompt for the C shell:

```
%
```

If you use the **-program** option frequently, you can save time b including the command in a startup (batch) file, and then simpl typing the name of the startup file at the DOS prompt.

For example, you might include the command above in a startu file that starts DTERM and creates a BSD shell. We include th following command in a file named **st_dterm.bat**:

```
dterm -program "/bsd4.3/bin/start_csh"
```

Now execute the startup file from the DOS prompt:

```
st_dterm
```

When you do so, DTERM starts up, prompts you for your username and password, and then creates a BSD shell.

* * *

Chapter 7

Using the Domain System with DTERM

Chapter 6 introduced DTERM, the Domain/PCI Connection's terminal emulator. This chapter explains how to use the Domain system with DTERM; specifically, it tells you how to:

- Use shell commands
- Use Domain/OS pathnames
- Run DPSS/Mail
- Use the **pci_assist** utility to overcome differences between DOS and Domain/OS filename conventions
- Create and use script files

Using Shell Commands

Shell commands let you access the Domain/OS utilities. These utilities perform traditional computing operations such as copying files and directories, and running and compiling programs.

You use a shell command by typing the command at DTERM's prompt and pressing <RET>. For example, typing the **wd** command from an Aegis shell (or typing **pwd** from a UNIX* shell) displays the name of your current working directory:

```
$ wd
//shoe/mckinley
```

All the Domain/OS commands are available to you through DTERM, except those that require graphics or windowing support. The Domain/PCI Connection does *not* support the following Domain/OS commands:

/com/crpad
/com/edfont
/com/netmain
/com/vt100
/com/xdmc
/bin/graph

Refer to the *Aegis Command Reference*, the *BSD Command Reference*, or the *SysV Command Reference* for information about commands in each environment.

Starting a Shell from Domain/OS

Domain/OS has three environments that you can switch among by starting a shell. At SR10, to start a BSD, SysV, or Aegis shell from any other Domain/OS shell, use one of the following commands:

/bsd4.2/bin/start_csh	(for 4.2BSD)
/bsd4.3/bin/start_csh	(for 4.3BSD)
/sys5/bin/start_sh	(for SysV)
/com/sh	(for Aegis)

The specified environment must be running on the server node.

---

*UNIX is a registered trademark of AT&T in the USA and other countries.

Using Pathnames

As described in *Getting Started with Domain/OS*, use a pathname in command lines to describe the location of a directory or file.

A pathname is series of names separated by slashes that describe the path the operating system must take to get from some starting point to a destination directory or file. A pathname begins with the starting point's name, and includes every directory name between the starting point and the destination object. A pathname ends with the name of the destination object. The slash (/) that separates each name in a pathname is equivalent to the DOS backslash (\) pathname separator.

Remember that two slashes "//" indicate the Domain system's top-level directory (the root directory).

The following are examples of pathnames:

//shoe/mckinley	Pathname for the directory called **mckinley** on the node **shoe**
mckinley/abc	Pathname for the file **abc** in the directory called **mckinley**

Using DPSS/Mail

DPSS/Mail is an optional Domain software package that allows you to send and receive messages. DTERM supports the command-line version of DPSS/Mail.

This section summarizes the differences between DPSS/Mail and Domain/PCI's implementation of this package. For additional information about DPSS/Mail commands, consult your *DPSS/Mail User's Guide*.

If this package is installed on the server you're connecting to, start DPSS/Mail with the **mail** command:

```
$ mail
```

Your PC displays the : prompt to indicate that the DPSS/ Mail program is running:

```
:
```

If you are running DPSS/Mail under DTERM, the Domain environment you're in determines which command you use to send a message. If you are using Aegis, send your mail message by pressing <RET> and <CTRL> Z. If you are using BSD or SysV, send your mail message with <RET> and <CTRL> D.

Displaying Filenames and Directories

While a filename or directory name on the Domain/OS system can have up to 32 characters (pre-SR10) or up to 255 characters (SR10), DOS file or directory names can have no more than eight characters and an optional extension of one to three characters. In addition, a DOS name cannot contain more than one period, and cannot contain blank spaces or any of these characters:

```
* + , : ; < = > ? [ ] | \ "
```

Because of these differences in naming conventions, DOS may be unable to recognize some names on the Domain system. You can display the filenames and directory names that DOS cannot see with the **pci_assist** utility.

Invoking the pci_assist Command

Invoke the **pci_assist** utility by entering the **pci_assist** command at a shell prompt.

Format

pci_assist *directory_name* [-a] [-i]

Arguments

directory_name	Indicates the name of the directory you want **pci_assist** to examine; if you omit this argument, **pci_assist** examines your current directory.

Options

-a	Displays the list of text file extensions listed in the **pci_cvrt.dat** file, if that file is in the directory being examined, or if it's in the **/sys/dpci** directory.
-i	Displays all the Domain/OS objects DOS cannot see (invisible objects).

If you do not specify any options, **pci_assist** displays the list of extensions in the **pci_cvrt.dat** file and all the Domain/OS names that DOS cannot display.

Example

This example shows the result of examining a Domain directory that contains four names that DOS cannot display.

```
$ pci_assist
```

```
PCI_ASSIST - Version 3.0
Server UASC file extension list from /sys/dpcc/pci_cvrt.dat:
      asm        bak        bat        c          dat
      doc        ftn        h          lst        map
      pas        txt        $$$
Domain objects with NO EQUIVALENT DOS name:
Domain name                          DOS name
-----------                          --------
monthly_reports                      None
user_data                            None
file.abc.bak                         None
ab++c                                None
```

DOS cannot display **monthly_reports** or **user_data** because their names have more than eight characters. DOS cannot display **file.abc.bak** because it has more than one period. It cannot display **ab++c** because it contains characters that are illegal in DOS.

Getting Access to "Invisible" Files

To provide DOS access to invisible Domain files and directories, you can do one of the following at a shell prompt:

- Change the Domain names. For example, you might shorten the name of the directory **monthly_reports** to **monthly.dir**:

  $ chn monthly_reports monthly.dir (Aegis environment)

- Create links to the Domain names with names that DOS can display. For example, you might create a link with the name **usrdata** that resolves to the directory named **user_data**.

  $ crl usrdata user_data (Aegis environment)

  For information on creating links, refer to the appropriate command reference for the environment (BSD, SysV, or Aegis) running on the server.

Creating and Using Script Files

DTERM lets you create script files, which you can use in place of the keyboard for input.

DTERM must be nonresident when you create or use a script file. DTERM is nonresident when you start it, but becomes resident the first time you use the Hot Key feature. When DTERM is resident, DTERM displays **RESIDENT** in the status line:

```
RESIDENT
```

If you want to create or use a script file and DTERM *is* resident, exit from DTERM (with the **logout** command) and then enter DTERM again.

Because script files contain control sequences, you cannot modify them with a text editor.

> **NOTE:** Do not use the Hot Key feature when you are creating or using a script file.

Creating Script Files

After you have made sure that DTERM is nonresident, follow this procedure to save keyboard output in a script file:

1. Press <ALT>O (the letter, not the number zero). DTERM displays a screen where you enter the name of the file where you want the output to be saved.

   In this screen, DTERM displays the name of the most recently created script file, if such a file exists.

2. If a script file already exists and you want to use it, press <RET>.

   Otherwise, enter the name of the file where you want the output saved and press <RET>.

If you enter only a filename, DTERM places the file in the directory from which you invoked DTERM. If you want the file in a different directory, enter a pathname that specifies the different location; for example, **d:\mynode\myfile**.

3. If the filename you enter already exists, DTERM prompts you for whether you want to overwrite the existing file, append to it, or specify a new file. Type **o** to overwrite, **a** to append, or **n** to specify a new file. If you choose **n**, then specify the name of a new file.

4. DTERM returns to the regular DTERM display, and shows the following in the status line to show that keyboard output is to be saved in a file:

```
OUT
```

5. DTERM writes everything you type at the keyboard into the specified file. DTERM continues to echo your keyboard input on the PC's screen.

6. When you are finished, close the file by pressing <ALT>O again. DTERM erases the display of **OUT** in the status line.

Using Script Files

After you have made sure that DTERM is nonresident, follow thi procedure to use the contents of a script file in place of keyboar input:

1. Press <ALT>I. DTERM displays a screen where you ente the name of the script file you want to use.

2. DTERM displays the name of the most recently create script file on the screen, if such a file exists. If you want t use that file, press <RET>.

   Otherwise, enter the name of the file and press <RET>. I the file does not exist, DTERM displays an error message enter a new filename and press <RET>.

3. DTERM displays the following in the status line to show that a script file is to be used for input rather than the keyboard:

```
IN
```

4. DTERM takes all keystrokes from the file as if you were typing them at the keyboard.

5. When DTERM has read all keystrokes from the file, DTERM erases the display of **IN** in the status line and returns to taking input from the keyboard.

* * *

Chapter 8

Using VT100 Applications with DTERM

Chapter 6 introduced basic DTERM capabilities. This chapter describes how you can use DTERM to run applications that take advantage of the VT100's special features such as the Advanced Video Option character attributes and the special graphics character set.

This chapter tells you about:

- Invoking an application
- Character attribute support
- Graphics character support
- Keyboard keys
- Modifying screen characteristics

Invoking an Application

To invoke a VT100 application, simply type the name of the application or the pathname at the prompt.

Format

```
$  application_name
```

Example

For example, to invoke the **vi** editor, start a BSD or SysV shell (if there isn't one already). Then enter the name of the application at the shell prompt.

```
/bsd4.3/bin/start_csh
% vi
```

Character Attribute Support

DTERM supports the VT100 terminal's Advanced Video Option character attributes. Any character on the PC screen can be highlighted with **bold** (bright), **blink**, **underline**, **reverse video**, or any combination of these.

Because of PC screen limitations, a color PC monitor displays the monochrome PC monitor underline attribute with the blue attribute.

Graphics Character Set Support

DTERM supports the 32 characters in the VT100 special graphics character set. For most of the characters in this set, the VT100 and PC representations of the characters are the same.

Some of the characters do not have an exact character match on the PC. DTERM maps each of the VT100's horizontal scanlines (1,3,5,7,9) to the PC's single horizontal line, which is equivalent to the VT100's scanline 5.

Table 8–1 shows the other VT100 characters that do not have an exact PC match and their PC equivalents.

Table 8–1. VT100 Graphics Character Equivalences

Character Name	VT100 Representation	PC Equivalent
Horizontal tab	␉	►
Formfeed	␌	ƒ
Carriage return	␍	Ç
Linefeed	␊	≡
Newline	␤	Ñ
Vertical tab	␋	¥
Not equal to	≠	╪

PC Equivalents for VT100 Keyboard Keys

Some VT100-based applications may require keys that are not o the PC keyboard. For example, an application might use th VT100 terminal's PF1 key since the PC doesn't have a PF1 key f providing help about the application.

To handle these differences, DTERM provides certain keys an key sequences to match VT100 keys. First, the PC's numeric ke pad keys (when NUMLOCK is *on*) are equivalent to the VT100' numeric keypad keys. Table 8-2 shows the other PC keys that ar equivalent to VT100 keys.

Table 8-2. VT100 Keys on the PC

VT100 Key	PC Equivalent
SETUP	<ALT> S
NO SCROLL	CTRL/S
PF1	F1
PF2	F2
PF3	F3
PF4	F4
LINE FEED	<ALT> <RETURN>
←	<ALT> - *or* ←
→	<ALT> = *or* →
↑	<ALT> 9 *or* ↑
↓	<ALT> 0 *or* ↓
ENTER	+ (numeric keypad)
, (numeric keypad)	. (numeric keypad)

Setting Screen Characteristics

The VT100 terminal lets you set up or alter certain terminal characteristics.

Two **set-up modes**, SET-UP A and SET-UP B, let you modify the terminal settings.

SET-UP A lets you:

- Change tab settings
- Move to SET-UP B

SET-UP B lets you:

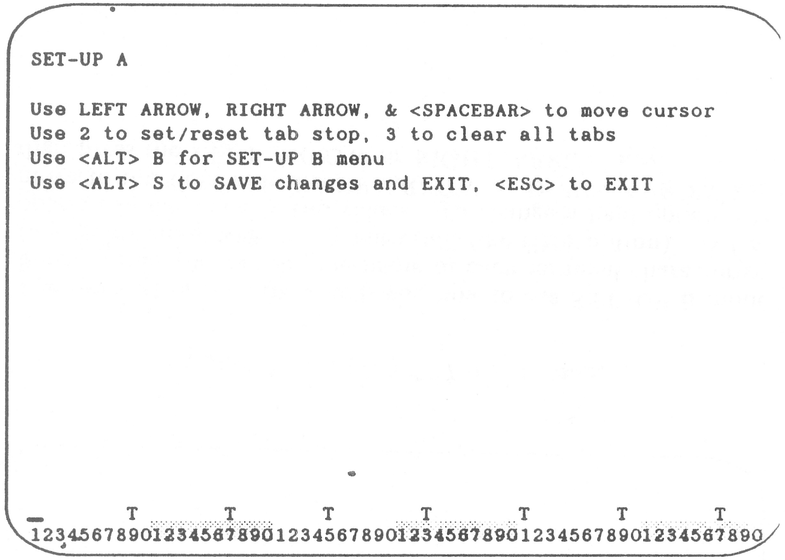
- Make the cursor a block or underline
- Toggle automatic linefeed
- Toggle the margin bell
- Toggle local echo

NOTE: Do not use the Hot Key feature when you are in either of the set-up modes.

Within the set-up screens, you can use the arrow keys: LEFT, RIGHT, UP, and DOWN. You can use the VT100 arrow keys for this movement (their representation on the PC is shown in Table 7-1) or the PC's numeric keypad arrow keys.

Using SET-UP A Mode

SET-UP A mode lets you change tab stops and move to SET-UP B mode. To enter SET-UP A, press <ALT>S. When you do so, your PC displays the screen shown in Figure 8-1. The top part of the screen tells you how to use SET-UP A mode.



```
SET-UP A

Use LEFT ARROW, RIGHT ARROW, & <SPACEBAR> to move cursor
Use 2 to set/reset tab stop, 3 to clear all tabs
Use <ALT> B for SET-UP B menu
Use <ALT> S to SAVE changes and EXIT, <ESC> to EXIT




_         T         T         T         T         T         T         T
1234567890123456789012345678901234567890123456789012345678901234567890
```

Figure 8-1. VT100 SET-UP A Mode

Example

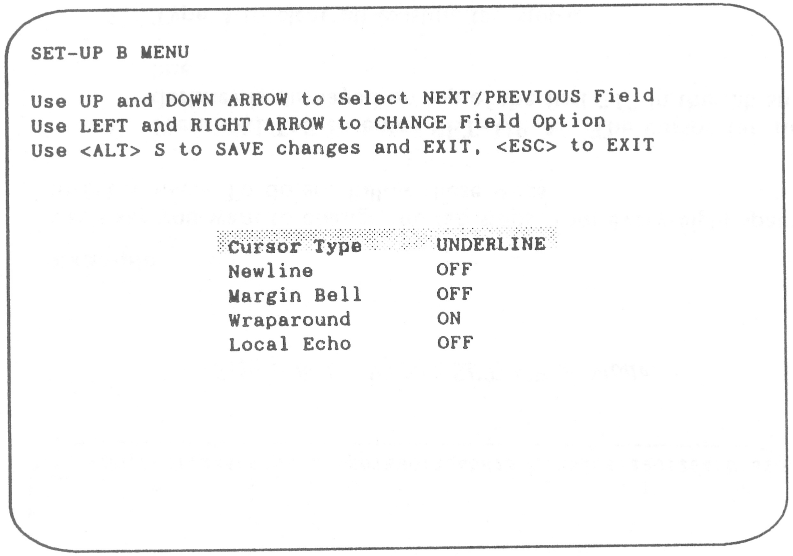
Let's say you want to change the tab stops from every eight space to every five. To do so, follow these steps:

1. Press <ALT>S to enter SET-UP A. The cursor (an underline) is displayed on top of the first "1" in the tab stop line.
2. Type **3** to clear all existing tab stops.
3. Move the cursor with the RIGHT ARROW key on top o the first 5 in the tab stop line.
4. Type **2** to set the tab stop.
5. Move the cursor on top of the first 0 in the tab stop line
6. Type **2** to set the second tab stop.
7. Move the cursor on top of the second 5 in the tab stop line.
8. Type **2** to set the third tab stop.
9. Continue moving the cursor and setting tab stops every five spaces until the end of the line.
10. If you want to save your changes and exit from SET-UP A, press <ALT>S. If you want to exit without saving your changes, press <ESC>.

Now the tab stops are set for every five spaces. The next time you enter SET-UP A, you'll see tab stops every five spaces rather than every eight.

Using SET–UP B Mode

SET–UP B mode lets you alter characteristics of your terminal. To enter SET–UP B, press <ALT>B from SET–UP A mode. When you do so, your PC displays the screen shown in Figure 8–2.



```
SET-UP B MENU

Use UP and DOWN ARROW to Select NEXT/PREVIOUS Field
Use LEFT and RIGHT ARROW to CHANGE Field Option
Use <ALT> S to SAVE changes and EXIT, <ESC> to EXIT


                Cursor Type       UNDERLINE
                Newline           OFF
                Margin Bell       OFF
                Wraparound        ON
                Local Echo        OFF
```

Figure 8–2. VT100 SET–UP B Mode

The top part of the screen tells you how to use SET–UP B mode Beneath that you can see the name of each terminal characteristi (the **field**) along with its current value (the **field option**). A fiel option can have one of two values. To change a field option, firs move to the field with the UP ARROW or DOWN ARROW key then press the LEFT ARROW or RIGHT ARROW key.

You can change these terminal characteristics with SET–UP B:

Cursor Type	The cursor can be a blinking **underline** or blinking **block**. The default setting is **underline**.
Newline	If **Newline** is **OFF**, <RETURN> generates a line-feed code (<LF>). If **Newline** is **ON**, <RETURN> generates a carriage return code followed by a line-feed code (<CR><LF>). **Newline** is **OFF** by default.
Margin Bell	If **Margin Bell** is **ON**, the PC bell sounds whenever the cursor is eight characters from the end of the current line. If **Margin Bell** is **OFF**, the PC bell does not sound. **Margin Bell** is **OFF** by default.
Wraparound	If **Wraparound** is **ON**, the eighty–first character of the current line is placed in the first character position of the next line. If **Wraparound** is **OFF**, the eighty–first character and all subsequent characters are overwritten in the last character position of the current line. **Wraparound** is **ON** by default.
Local Echo	If **Local Echo** is **ON**, the characters entered at the PC keyboard are echoed to the screen and sent to the server node. If **Local Echo** is **OFF**, typed characters are sent only to the server. **Local Echo** is **OFF** by default.

Example

Let's say you want to turn the Margin Bell ON. To do so, perform these steps:

1. Move to the Margin Bell field with the DOWN ARROW key.
2. Select the field by pressing the RIGHT ARROW key.

   The field option changes from **OFF** to **ON**.
3. Press <ALT>S to save the change and return to VT100 emulation mode.

Chapter 9

Managing the Domain/PCI Connection Server

This chapter explains to the Domain/PCI system administrator how to manage the Domain/PCI Connection server. The server is the software component of the Domain/PCI Connection that runs on the server node, giving the PC access to the Domain system. To use the information in this chapter you should be familiar with the Domain system.

In this chapter, you'll learn the tasks associated with managing the Domain/PCI Connection server. The tasks are

- Setting up the offers file
- Setting up the server configuration file
- Starting the Domain/PCI Connection server
- Stopping the Domain/PCI Connection server.

Before You Begin

This section lists which manuals you, the system administrator, should read prior to reading this manual. Typically, any task that can be done from the PC is the user's responsibility and any task that can done from the server node is your responsibility. To help you delegate these tasks, you should perform all tasks in the *System Administrator's Guide* of this manual.

Be sure you've read the following documents (in the order in which they are listed) completely *before* proceeding with this chapter. This is important because Domain/PCI server software and PC hardware installation information is not covered in this manual:

- *Domain/PCI Connection Release Document*, which discusses how to install the software on the server and describes client and server software bugs and limitation information.
- *Planning for and Installing the Domain/PCI Connection*, which describes how to install and run diagnostic tests on product–specific Domain/PCI Connection hardware.
- Any documentation, such as installation guides and addenda, that comes with connection hardware products from vendors other than Apollo Computer, such as the 3Com EtherLink II adapter.

You must perform the server installation steps described in the release document and complete the procedures in this chapter before the user can install the client software and start the Domain/PCI Connection on the PC.

Setting Up the Offers File

Before you start the Domain/PCI Connection server for the first time, you must set up the **offers file** and configure the Domain/PCI Connection server.

Offers are the Domain system resources that users can connect to from their PCs. You must set up an offers file in order to connect to a printer. The offers file is used only for printer connections.

However, with versions 1.0 or 2.0 of Domain/PCI SIO connection products, you must also set up an offers file to access directories if users use the **net use** command. If users want to use the **net use** command, once they have connected to an offer (see Chapter 4), they can use the Domain resource associated with the offer.

The offers are defined in the **pci_cmd.dat** file, which is located on the server node in the **/sys/dpci** directory.

The **pci_cmd.dat** file contains a list of **offer definitions** for Domain directories and Domain printers. You can specify up to 128 offer definitions in the **pci_cmd.dat** file.

This section describes:

- The syntax for defining Domain **printer** offers
- The syntax for defining Domain **directory** offers
- The initial offers file supplied with your Domain/PCI Connection, which you can modify if necessary
- How to create offer definitions and add them to the offers file

NOTE: If you are concerned about security, be sure to keep restrictive permission rights on the offers file.

Setting Up Domain Printer Offers

The Domain/PCI Connection lets you print files on standard Domain printers, including those print devices that support the PostScript page description language.

All Domain/PCI Connection products must have printer offers defined in the **pci.cmd.dat** file *before* the user attempts to print a file on a Domain printer. This section first describes the syntax for defining printer offers, and then shows you how to define offers for different types of files.

Format

offer *offer_name*= **-pr** *printer_name* [**-pcprint**]
[**-transparent**][**-font** *fontname*][*prf_option*...]
\char [**\not[ranslate]**] [**\setup** *pathname*] [**\reset** *pathname*]

Arguments

offer_name

Specifies the name for the offered printer. The name can be any combination of characters with a maximum of eight characters. This name is associated with the *printerdevice* argument when the user connects to the printer with the **net connect** or **net use** commands. The format of the **net connect** command that associates a printer device (such as lpt2) with an offer is

net connect *printerdevice*: *offer_name*

-pr *printer_name*

Specifies the name of the printer. This name must be the same as the name specified with the **-n** option on the **prsvr** command, which starts the print server for this printer (the **prsvr** command is described in the *Aegis Command Reference*).

You can use the same printer for any number of offers. For example, you might define two offers for a particular printer: one for printing files from a PC application that generates files already in PostScript format, and one for printing all other files.

\char

Indicates that the offer is a printer device.

Options

-pcprint

Indicates that files will be converted to PostScript format before being sent to the printer. When you include this option, you must also include the **-transparent** option.

-transparent

Indicates that files are to be sent to the printer without any additional formatting.

Include this option in two cases:

- When you are defining an offer definition for printing files that are already in PostScript format
- When you are defining an offer that uses the **-pcprint** option

-font *fontname*

Specifies the font to be used for printing when you include the **-pcprint** option.

fontname can be one of the following:

Courier
Times-Roman
Helvetica

If you do not specify **-font**, the Domain/PCI Connection uses Courier.

prf_option

Specifies one or more of the options available with the **prf** command, described in the *Aegis Command Reference*.

\not[ranslate] Prevents file type conversion from PC text file to Domain uasc file type.

\setup *pathname*
Specifies a file that contains printer control codes used to select a specific printer format, such as PostScript, HP LaserJet, and HP Plot. The server places this code at the top of the file to print. This option is used to select a specific printer format on printers that support multiple printer formats.

\reset *pathname*
Same as the **\setup** option. This option is typically used after a file has printed to set the printer to another printer format. This option can be used with printers that support multiple printer formats.

Printing Text Files

To establish an offer to be used for printing standard text files, create an offer definition using this format:

offer *offer_name*= **-pr** *printer_name* **\char**

For example, this offer definition establishes an offer called **prspin** for the standard Domain printer called **spin**:

```
offer prspin= -pr spin \char
```

Printing Text Files without File Type Conversion

Typically, PC text files that are printed on a Domain printer are converted to uasc format (with linefeed terminators). To print text files without converting the files to uasc format, create an offer definition using this format:

offer *offer_name*= **-pr** *printer_name* **\char \notranslate**

For example, this offer definition establishes an offer called **prprnt** for the standard Domain printer called **spin** without performing file type conversion:

```
offer prprnt= -pr spin \char  \notranslate
```

Printing Files that Contain PC Graphics

You can print files that contain PC graphics on any Domain printer that supports the PostScript page description language.

To establish an offer to be used for printing files that may contain PC graphics, create an offer definition using this syntax:

offer *offer_name*= **-pr** *printer_name* **-pcprint -transparent [-font** *fontname*][*prf_option...*] **\char**

The arguments and options are described in the section "Setting Up Domain Printer Offers" earlier in this chapter.

You can also use this offer for printing files that contain only text.

For example, this offer definition establishes an offer named **laser** for the printer called **apple,** which you can use for printing files that contain PC graphics as well as text:

```
offer laser= -pr apple -pcprint -transparent \char
```

NOTE: If your application has a printer configuration utility, be sure to select **IBM Graphics printer** as the printer choice. This selection enables the server to properly convert PC graphics characters to PostScript format.

Printing Files Already in PostScript Format

To establish an offer to be used for printing files that are already in PostScript format, create an offer definition using this format:

offer *offer_name*= **-pr** *printer_name* **-transparent \char**

For example, this offer definition establishes an offer named **laspr** for the printer called **apple**, which you can use for printing files that are already formatted with PostScript:

```
offer laspr= -pr apple -transparent \char
```

As you can see, this offer uses the same printer, **apple**, as the previous example.

> **NOTE:** If your application prints documents on a PostScript device, be sure that the application has the printer configured for **letter-size** paper.

Printing Files on Printers that Support Multiple Printer Formats

To establish an offer to be used with a printer that supports multiple printer formats, such as PostScript, HP LaserJet, and HP Plot format, create an offer definition using this format:

offer *offer_name*= **-pr** *printer_name* **\char \setup** *pathname* **\reset** *pathname*

For example, this offer definition establishes an offer named **laserpr** for the printer called **lexicon**, which you can use for printing files on a printer that allows multiple printer formats:

```
offer laserpr= -pr lexicon \char \setup //wayside/
hpjetcode \reset //wayside/pscode
```

This offer sets up the printer **lexicon** to print a file using HP LaserJet printer format. When the file has completed printing, the printer is then reset to PostScript printer format.

Setting Up Directory Offers (Versions 1.0 and 2.0)

Domain directory offers are used only to support the **net use** command for versions 1.0 and 2.0 Domain/PCI SIO connection products.

Format

offer *offer_name*= *pathname* **\block** [*password*]

Arguments

offer_name=	Specifies the name for the offered directory. The name can be any combination of characters with a maximum of eight characters. This name is used as the *offer_name* argument when you connect to the offer with the **net use** command (see Chapter 5).
pathname	Specifies the complete Domain pathname of the directory being offered.
\block	Indicates that the offer is a directory.

Options

password	Specifies the directory's password. If a directory has a password, only users who know the password can access files in that directory. The password can be any combination of characters with a maximum of eight characters. Do not include spaces in passwords.

For example, this offer definition establishes an offer named **accounts** for a Domain directory whose pathname is **//sales/accounts:**

```
offer accounts= //sales/accounts \block
```

The Initial Offers File

Your Domain/PCI Connection comes with an offers file called **/sys/dpci/pci_cmd.dat**. You can modify this file or use it as is.

The file contains one offer definition:

```
OFFER PRINT= -PR P \CHAR
```

This definition establishes an offer called **print**, which makes **p**, the default Domain system printer, available to your PC. The default Domain printer is the one specified with the **/sys/print** directory or link on the server node.

Creating New Offer Definitions

You can add new offers to the offers file. In this section, we show you how to edit the offers file. Also, we illustrate why you don't need to use offers for directories if you use the **net connect** command.

Editing the Offers File

You can edit the **pci_cmd.dat** file on the server node at any time. *When you do so, make sure to restart the Domain/PCI Connection server so that your changes can take effect.*

When you edit the **pci_cmd.dat** file you need to:

- Notify the appropriate PC users of changes to available offers.

  or

- Edit the client startup file on the PC to include **net connect** or **net use** commands that connect to the new offers and distribute this file to the appropriate PC users.

Using net connect Instead of net use

The **net connect** command makes connections much easier than the **net use** command. First, let's look at what you need to do if you use the **net use** command. The following offer definition in a **pci_cmd.dat** file creates an offer called **today** for the **accounts** directory on the node called **sales**. It also provides the password **sesame** for the offer.

```
offer today= //sales/accounts \block sesame
```

Now, the **net use** command connects your PC's drive named E to this new offer:

```
C:> net use e: \\domain\today sesame
```

If you use the **net connect** command to make the same connection to the \\sales\account directory, it can be made directly from the command line without an offer:

```
C:> net connect e: \\sales\accounts
```

The Domain/PCI Connection will automatically connect drive E to this offer if you add the command to the PC client startup file.

Configuring the Server

If you are not using a Domain/PCI direct connection product, read the section "Starting the Server for Domain/PCI-1."

When the Domain/PCI Connection server starts up, it searches for a server configuration file. It first looks in '**node_data/dpcinet/pci_cfg.dat** on the server node. If the server configuration file is not there, it then looks in **/sys/dpcinet/pci_cfg.dat** for a list of node IDs of the PCs that are allowed to connect to the server.

The Domain/PCI Connection supplies a **pci_cfg.dat** file that you can modify if necessary. However, if you make a change to the **pci_cfg.dat** file, you must restart the server in order for the changes to take effect. Starting the server is described in this chapter.

The following is an example of a **/sys/dpcinet/pci_cfg.dat** file:

```
# DPCI configuration file
#
# By default, Allow up to 50 clients simultaneously
maxclients=3
#
# A null list below means we allow any PC on the network
# to connect to this server
#pcs in Corey's office
8700f
11630
#pcs in Beth's office
18dfc
6385f
562f7
```

In this example, although we've listed five legal candidates for the Domain/PCI Connection, **maxclients=3** limits the connections on the server to a maximum of three connections at any given time.

Format

maxclients=*n*

#

pc_node_id

Arguments

n	Specifies the number of PCs that can be simultaneously connected to the server. You can specify a value between 1 and 50. If you do not specify a **maxclients** line, the default is 50.
pc_node_id	Specifies the hexadecimal node ID of a PC that is allowed to connect to the server. If you do not specify any *pc_node_ids* in the **pci_cfg.dat** file, then any PC can connect to the server (assuming the **maxclients** limit has not been exceeded).

Options

#	Indicates a comment line.

Starting the Server for Direct Connection Products

After you set up the offers file, **pci_cmd.dat**, and the configuration file, **pci_cfg.dat**, you can start the server, **dpcinet**, for any of the Domain/PCI direct connection products.

Format

cps /sys/dpcinet/session [**-retries** *zz*] [**-pingtime** *n*]

Options

-retries *zz*	Specifies the number of times the server should try to send a packet again after encountering an error. The maximum number of retries is 250. The default number is 60.
-pingtime *n*	Specifies the interval of time at which the server verifies whether or not a connected PC is active. If the PC does not respond to the **pingtime inquiry**, which is the time interval that you specify, the inactive PC loses the resources it is using to the next PC that is connected. The major resources include the server and the DTERM process. The interval *n* is expressed in minutes. The valid entries are from 0 to 60. The default number is 0.

Starting the Server for Domain/PCI-1

After you set up the offers file, **pci_cmd.dat**, you can start the server, **dpci1**, for Domain/PCI-1. The Domain/PCI-1 connection is configured on the command line.

Format

```
cps  /sys/dpci1/session [-line x] [-baud  yyyy] [-retries zz]
```

Options

-line *x*	Specifies either the physical or logical SIO port number on the server node, where *x* is the port number. The number does not need to be the same as that on the PC side. The default is line 1. The following table lists the default maximum physical number of SIO ports available on specific Domain nodes:

Node	Maximum Physical SIO Ports
DN3000	1
DN4000	3
DN300	2

For more information on physical and logical SIO ports see "Using Serial Expansion Units" in this chapter.

-baud *yyyy*	Specifies the baud rate, where *yyyy* is the rate. The value of *yyyy* can be 300, 600, 1200, 2400, 4800, or 9600. The baud rate must match the baud rate used on the asynchronous communications port of the attached PC. The default is 9600 baud.
-retries *zz*	Specifies the number of times the server should try to send a packet again after encountering an error. The maximum number of retries is 250. The default number is 10.

Using Serial Expansion Units

Read this section only if you are using Domain/PCI-1.

Serial expansion units (SEUs) are boards that enable you to add additional SIO ports to your node. As described in the **-line** option of the **/sys/dpci1** command, each Apollo node comes with a specific number of **physical SIO ports**. For each physical port that comes with your node, the **‘node_data/dev/sio** directory for the node contains an object file called **sio***x* (where *x* in the SIO port number).

The ports you add by using an SEU can be thought of as **logical SIO ports.** Some examples of SEUs are the Danford board, the SPE board, and the EMT board. This section briefly describes how to use SEU boards with a Domain/PCI SIO product. For detailed information about these boards, consult the product's documentation.

Typically, if you are not using SEUs, you are limited to the number of physical SIO port that are built into your node. Thus, for example, when you start the Domain/PCI-1 server on a DN4000 workstation, you can specify either SIO ports 1, 2, or 3 in the **-line** option.

However, if you have an SEU in your server, you can specify a **-line** option up to SIO 50 as long as you perform the following tasks:

1. Install the SEU hardware and software as described in the manual that comes with the product. For example, the Danford board requires that you run **ssiomonit**. This software creates objects in **/dev** called **ssio***x*.
2. Run any diagnostic software that is needed to test the product's installation.
3. To use these **ssio***x* objects, create a link from the /dev directory to the SEU object **ssio***x*

   ```
   crl /dev/sio4 ssio1
   ```

   This example links SIO port 4 to the first port (ssio1) on the board.

4. Type the following command to use the SIO port

   /sys/dpci1/session -line 4

The Server Processes Domain/PCI Creates

The server startup procedure initially creates two background processes on the server node that allows the Domain/PCI Connection to run as long as the server node is running. You can see the processes if you type the Aegis **pst** command or the UNIX command **ps -aux.**

```
$ pst
-----------------------------------------------------------------
Node:  18DFC
Time:  Wednesday, April 6, 1988   4:13:06 pm (EDT)
------------------------------------------------------------ Proc-
essor | PRIORITY |  Program | State |    Process Name
 Time (sec)| mn/cu/mx |  Counter |         |
-----------------------------------------------------------------
  10608.912  -- -- --   --------   -----   <Null Process>
    297.194  -- -- --   --------   -----   <Aegis Processes>
    392.329  16/16/16    32163AA   Wait    display_manager
      0.694   3/14/14    3215EAA   Wait    mbx_helper
      2.184   3/13/14    32163AA   Wait    srvr_process_mngr
      2.080   3/13/14    32163AA   Wait    mail
     18.066   3/14/14    32163AA   Wait    alarm_server
      5.542   3/11/14    32163AA   Wait    process_10
      1.759   3/14/14    32163AA   Wait    dpcinet_session
      1.698   3/14/14    32163AA   Wait   uid=3b4e2e0f.18dfc
-----------
  22401.838
```

In this example, the last two processes (dpcinet_session and an unconnected server process, uid=3b4e2e0f.c8dfc) are created when you start up a Domain/PCI direct connection product server. If you are running Domain/PCI-1, you see a dpci1_session_sio*x* process displayed (where *x* is the number you entered in the **-line** option) and an unconnected server process. When the first PC connects to the server, the unconnected server process is replaced with a specific server process that identifies the PC's node ID and a new unconnected server process is created. For example,

```
10.591    3/14/14    32163AA    Wait   dpcinet_server_fed5a
1.398   3/13/14 32163AA   Wait    uid =3b51e482.30008dfc
```

Here a PC with node ID fed5a connects to the server and a new unconnected process is created in anticipation of another PC connecting to the server. If a PC invokes DTERM, the server creates one new process, a specific DTERM process as shown below

```
1.466    3/13/14     32163AA    Wait   dpcinet_dterm_fed5a
```

It does not use the existing unconnected server process created when the last PC connected. As subsequent PCs invoke and log out of DTERM, the number of specific server processes and unconnected server processes increases and decreases respectively. This also occurs as subsequent PCs connect and disconnect from the server. Domain/OS enables you to create up to 56 user processes. If the server reaches its process limit, the next PC user who attempts to connect will get a following error message:

```
Remote server listen not posted.
```

For more information on creating a process server, see the *Using Your Aegis Environment*, *Using Your BSD Environment*, or *Using Your SysV Environment* and the command reference appropriate for the environment you're running (Aegis, SysV, BSD).

Stopping the Server

To stop the process used by the Domain/PCI Connection server and close the server, enter the Aegis **sigp** command or the UNIX **kill** command.

Format

sigp *productname*_**session** [–s –l] (Aegis environment)

kill *pid* (UNIX environment)

Arguments

productname	**dpcinet** for Domain/PCI direct connection product or **dpci1** for a Domain/PCI SIO connection product. (Aegis environment only)
pid	The process ID of the *productname*_session. (UNIX environment only)

Options

–s[top]	Stops the entire process cleanly (closing streams, etc.)
–l	Lists processes signaled.

Example

To stop all Domain/PCI direct connection products servers on a node, type

```
sigp dpcinet_session –s –l
```

To stop a specific server, type

sigp dpcinet_server_*node_id*

where *node_id* is the node ID of the PC.

To stop the a Domain/PCI SIO connection product server, for example, type

```
sigp dpci1_session_sio2 –s –l
```

* * *

Chapter 10

Using Permission Rights

As you saw in Chapter 9, when you start the Domain/PCI Connection, it creates a **server process**, which handles communication between each connected PC and the Domain system. The Domain/PCI Connection can create up to 50 server processes, one for each user.

The Domain/PCI Connection server must have access to certain files and directories on the server node in order to operate. The Domain/OS controls permission to its files with a system protection mechanism for each environment. For example, the Aegis protection mechanism is called an **access control list (ACL)**. Every file and directory in the Domain system has an access control list that defines who can use the object and what operations users can perform on that object.

This chapter explains

- The server's subject identifier (SID)
- Server initialization
- Reports of protection errors in the server's error log file
- Locked objects when protection rights are correct
- How DOS interprets protection errors

This chapter assumes that you are already familiar with permission rights for the environment you're using; if you need more information on permission rights, please see *Using Your Aegis Environment*, *Using Your BSD Environment*, or *Using Your SysV Environment* and the command reference appropriate for your environment.

The Server's Subject Identifier (SID)

The Server's Subject Identifier (SID) identifies which users have specified permission rights. When a PC connects to the server node, a server process is created. If a PC uses the **dterm** command, a DTERM process is created on the server node. Each device that the PC connects to has a SID associated to it. However, the DTERM process contains a single SID.

Generally a user's SIDs are identical and are set by the **net login** command. They are obtained from the Domain/OS registry entry associated with the username and password. However, some exceptions are

- For Domain/PCI SIO connection products, if a device_name was connected using the **net use** command, the SID is *user.none.none*.
- If a device_name was connected using **net connect** with the **-login** option, the SIDs for that device_name are set from the **-login**.
- If you invoke DTERM with the **-login** override option, the SIDs for the DTERM process are set to those of the **-login**.

If you exit from DTERM with the <SHIFT> <ESC> method, you terminate the DTERM process on the server node; however, this does not affect any of the SIDs of the device_name in the server process.

If you use **net logout** or reboot your PC and reconnect, you break all device_name connections. In these cases, you will need to use **net login** to set new SIDs.

NOTE: If you turn off your PC, or reboot without reconnecting, the server process itself does not go away unless you are using the **-pingtime** option, which is described in Chapter 9.

The 'node_data Directory

When it starts up, the server process deletes and creates files in the **'node_data** directory. The names of these files vary, depending on whether you are running a Domain/PCI SIO connection product or a Domain/PCI direct connection product, depending on what the server's SIO line number is, and how many server processes are present.

For a given filename, such as **dpci***product***_x_server***y*, the following conventions are used:

product
Indicates the name of the Domain/PCI Connection product you are using. (See Table 1-1 for the values of *product*.)

x
If you are using a Domain/PCI direct connection product, this remains **x**. If you are using a Domain/PCI SIO connection product, this indicates the same number displayed in variable *y* below.

y
If you are using a Domain/PCI direct connection product, this indicates the number of server processes minus one; that is, if this is the second server process created, *y* will be 1 (*y* can be a number from 0 to 49). If you are using a Domain/PCI SIO connection product, this indicates the SIO line number you entered on the **-line** option when you started the server.

The server performs the following steps upon startup:

1. Deletes the file **'node_data/dpci***product***_x_server***y* if it exists. The file does not exist if this is the first time you are starting up the server.

2. Creates the file '**node_data/dpci**_product_**_x_servery**. The server uses this file to share data between the originating process on the server node, **/sys/dpci**_product_**/session**, and any server processes.
3. Creates '**node_data/dpci**_product_**_xpombxy** the mailbox file, which is used for communication between processes.
4. Creates the spool directory '**node_data/dpci_queue** if it does not already exist. The Domain/PCI Connection creates files in this directory for buffering parts of files during printing.

To perform these steps, the server must have permission rights to the '**node_data** directory, and must have delete, read, and write protection to the files in '**node_data**. Specifically the following items must provide the minimum protection:

- The permission rights on the '**node_data** directory.
- The initial (default) permission rights for directories created under the '**node_data** directory.
- The initial (default) permission rights for files created under the '**node_data** directory

You should check the permission rights, initial directory permissions rights, and initial file permission rights for the '**node_data** directory. If the permission rights are incorrect, you should change them to meet the minimum protection requirements.

The Error Log File

The Domain/PCI Connection server writes error messages to the file '**node_data/dpci_error_log** if it encounters any protection errors. Like the other files in the '**node_data** directory, the server must have write protection to **dpci_error_log.**

As described in the previous section, the server creates the file '**node_data/dpci**_product_**_x_servery** upon startup, after deleting any existing file with the same name. If the file already exists, but the server does not have delete rights to it, the server cannot create the new file because it cannot delete the old one; if this happens the server writes an error message to the error log file.

The following example shows a typical sequence of messages that can occur if the server cannot create the '**node_data/dpci**_product_**_x_servery** file. Each message is explained after the example.

```
[1]  Domain(R)/PCI(TM) SERVER   x.xx  28-FEB-88.
************************************************************


[2]  Entry created on  06-Mar-1988 10:04    Node id: 2469
Process uid: 2C54E666.20002469

Server initialization error:  Calling ms_$crmapl (). -
name already exists (OS/naming server)
************************************************************


[3]  Entry created on  06-Mar-1988 10:04     Node id: 2469
Process uid: 2C54E655.E0002469

SESSION: Critical error on server initialization.
************************************************************


[4]  Entry created on  06-Mar-1988 10:04     Node id: 2469
Process uid: 2C54E666.20002469

Server initialization error:  Unable to open mailbox. -
channel not open (library/MBX manager)
************************************************************


[5]  Entry created on  06-Mar-1988 10:04     Node id: 2469
Process uid: 2C54E666.20002469

Server closing down.
****************************************************
```

1. Shows that the server has started and indicates the server's version number (important for reporting software problems).

2. Indicates that the server was unable to create the file **'node_data/dpci***product***_x_servery** .

3. Indicates that the server process on the server node has recognized that the server had a critical error.

4. Indicates that the server received the error from the server node process.

5. Indicates that the server is stopping.

Locked Objects when Permission Rights Are Correct

As described earlier in "The 'node_data Directory," when a server starts up, it opens a number of files. If, for some reason, the server cannot exit cleanly (for example, if the server node crashes), these files may be left open, and thereby become locked. If you are unable to start the server, but know that the permission rights are correct, one or more files may be locked open.

To correct the problem, do the following from the server node:

1. Stop the Domain/PCI Connection server process on the server node with this Aegis command (or use the **kill** command in a UNIX environment):

   sigp dpci*product***_session[_sioy] –s**

   where *product* is the product name (as listed in Table 1–1 and *y* is the SIO line you specified when you invoked Domain/PCI SIO connection product.

2. Delete all the dpci files in the **'node_data** directory. For example:

   dlf 'node_data/dpci?*

   Note that locked objects cannot be deleted.

3. Unlock all dpci files remaining in the **'node_data** directory. (Repeat the command line below for each file, substituting the file's name for *filename*):

   ulkob 'node_data/*filename* **–f**

4. Delete the previously locked files:

   dlf 'node_data/dpci?*

5. Start the Domain/PCI Connection server on the server node:

   cps /sys/dpci*product***/session** [*arguments*]

How DOS Responds to Permission Violations

When you use the Domain/PCI Connection to access a Domain file or directory for which you don't have permission rights, the Domain/PCI Connection server returns an error message to DOS. However, because DOS does not have any permission control mechanism, it cannot issue specific permission errors. The typical DOS responses to permission violations are shown in the following examples:

- From a PC, you try to display the contents of a Domain directory for which you don't have permission rights:

  C:> **dir d:my_node\my_dir**

  DOS displays the following instead of displaying the contents of the directory:

```
MY_DIR <DIR>    date    time
1 File(s)    xxxxxxx bytes free
```

- You try to change directories, specifying a Domain directory for which you don't have permission rights:

```
C:> cd d:\my_node\my_dir
```

DOS responds with:

```
Invalid directory
```

- You try to copy files for which you don't have read permission:

```
C:> copy d:\my_node\my_dir\my_file a:
```

DOS responds with:

```
0 File(s) copied
```

NOTE: If DOS doesn't perform a command as expected, and the permission rights on objects are correct, make sure that the pathname you have entered is a name that DOS can recognize. Use the **pci_assist** utility, described in Chapter 7, to display filenames that DOS cannot display.

⁂

Chapter 11

System Administration Command Reference

As the system administrator, you need to use various commands to monitor and maintain the network. With the addition of PCs directly to the network, you need to understand how the Domain system administration commands display information about PCs on the network.

The commands presented in this chapter are the same shell commands used in any Domain/OS network. We assume that you are already familiar with Aegis shell commands. This chapter shows only information about a PC on the network. For more information about the commands presented in this chapter, refer to the *Aegis Command Reference*.

bldt — Display version of DOS (build_time)

Purpose

Displays the version of DOS and the version and date of the **dpci***product***\xport** software (from the PC's **\dpci\dpci***product***\xport.exe** file).

Format

bldt -n [*pc_node_id*]

Options

pc_node_id	Displays the build time of the PC whose *pc_node_id* is the hexadecimal ID (or node name) of the PC. You can obtain this ID by using the **netstat** command described in this chapter.

Arguments

-n	Displays information about the specified PC.

Example

```
$ bldt -n fed6b

    **** Node FED6B ****   FED6B
DOS 3.20 with DOMAIN(R)/PCI(TM) XPORT 4.00 - August 18, 1988
3:23:36 pm
```

```
$ bldt -n //blithesdale

    **** Node 29C15.FED6B ****    "//blithesdale"
DOS 3.20 with DOMAIN(R)/PCI(TM) XPORT 4.00 - August 18, 1988
3:23:36 pm
```

In the second example, we named the PC **blithesdale**. You can give your PC a node name with the **ctnode** command described later in this chapter.

NOTE: The number that precedes the node ID (in this case, 29C15), is the node address. This number identifies the network on which the PC resides. Be sure that the server node to which you connect the PC is on the same network.

ctnode — Catalog a PC in the network (catalog_node)

Purpose

Allows you to catalog your PC with a name. The name displays in the **bldt, lcnode, netstat,** and other network commands. From the PC, you cannot use a node's *node_name* to access the Domain/OS file system; you must use the hexadecimal node ID. To uncatalog a node, refer to the command **uctnode** later in this chapter.

Format

ctnode [*node_name pc_node_id*].

Options

node_name	Specifies the name of the PC that you want to catalog.
pc_node_id	Specifies the hexadecimal ID of the PC you want to catalog. You can obtain the ID by using the **netstat** command described in this chapter.

Example

```
$ ctnode blithesdale fed6b
```

This example names the PC fed6b **blithesdale**.

lcnode — List PCs and nodes connected to the network (list_connected_nodes)

Purpose

Lists the PCs (and nodes) currently connected to the network. The list contains the ID of every PC connected, the time at which the PC was started, the current time, and the name of each PC's entry directory. Also, **lcnode** shows the order of the PCs on the network.

Format

lcnode

Example

```
$ lcnode

The node ID of this node is 18DFC.
 185 other nodes responded.
Node ID     Boot time            Current time   Entry Directory
18DFC    1987/08/24  8:42:39   1987/08/24 10:34:53  //corey
1C9D3    1987/08/23 16:26:21   1987/08/24 10:43:11  //beth
fed6b  1980/01/04  1:00:18  1980/01/04  1:43:43  //blithesdale
1C98   1987/08/23 16:18:13   1987/08/24 10:44:50  //wayside
1700F    1987/08/24 11:35:00   1987/08/24 12:48:09  //kathy
                    .
                    .
                    .
```

netstat — Display network statistics (network_statistics)

Purpose

Writes a summary of network activity to standard output.

Format

netstat -n [*pc_node_id*] **-config -l -a**

Options

pc_node_id	Specifies the hexadecimal ID of the PC.

Arguments

-n	Displays information about the specified PC.
-config	Displays only PC-specific information.
-l	Provides more information than the summary.
-a	Lists information on *all* nodes in the network. You can use this to obtain a *pc_node_id*.

Example

$ netstat -n fed6b

```
    The net_ID.node_ID of this node is 29C15.18DFC.
**** Node FED6B ****   FED6B
Time 1980/01/04.04:03:54  Up since 1980/01/04.01:00:18
Net I/O:         total=  51126   rcvs =    427   xmits =  50699
Winchester I/O:  total=      0   reads=      0   writes=      0
```

$ netstat -n //blithesdale

```
    The net_ID.node_ID of this node is 29C15.18DFC.
**** Node FED6B ****   //blithesdale
Time 1980/01/04.04:03:44  Up since 1980/01/04.01:00:18
Net I/O:         total=  51077   rcvs =    421   xmits =  50656
Winchester I/O:  total=      0   reads=      0   writes=      0
```

The following example illustrates the **-config** and the **-l** options:

$ netstat -n //blithesdale -config -l

```
    The net_ID.node_ID of this node is 29C15.18DFC.
**** Node FED6B ****   //blithesdale
Time 1980/01/04.06:26:13  Up since 1980/01/04.01:00:18
Net I/O:         total= 131013   rcvs =   1247   xmits = 129766
  0 page-in  requests issued.
  0 page-out requests issued.
  0 page-in  requests serviced.
  0 page-out requests serviced.
Detected concurrency violations -- read: 0    write: 0
 Xmit count        1247          Rcv eor            1
 NACKs                0          Rcv crc           99
 WACKs               29          Rcv timout        84
 Token inserted       0          Rcv buserr         0
 Xmit overrun         0          Rcv overrun        1
 Xmit Ack par         0          Rcv xmit-err     586
 Xmit Bus error       0          Rcv Modem err    586
 Xmit timout          1          Rcv Pkt error      0
 Xmit Modem err       0          Rcv hdr chksum     0
 Xmit Pkt error       1          Rcv Ack par       81
    Delay switched OUT.
Winchester I/O:  total=      0   reads=      0   writes=      0
 Not ready            0          Contrlr busy       0
 Seek error           0          Equip check        0
 Drive time out       0          Overrun            0
 CRC error percentage: 0.00%
 Last ring hardware failure detected by node ED86
   on 1980/01/04 at 06:13
    Node was not receiving clock signals.
A total of 0 memory errors were detected.
  NODE CONFIGURATION    Node Type:  (id 255 unrecognized)
```

probenet — Probe network and display error statistics (probe_network)

Purpose

Broadcasts packets to the diagnostic socket in all PCs and nodes, then requests error counts indicating the status the broadcast was received with. It compiles counts from the topology list and reports them to standard output.

Format

probenet -n [*pc_node_id*]

Options

pc_node_id	Specifies the hexadecimal ID of the PC you want to probe.

Arguments

-n	Displays information about the specified PC.

Example

```
$ probenet -n fed6b

There are      1 nodes in the test.
Broadcasting 10 1024-byte packets 8/26 15:15:06  failures =  0
Last ring hardware failure detected by node ED086  at 14:23
                                   MODEM
NODE  NAME  ATTEMPT  ERRS   ERRS   BIPH    ESB  TOKENS=      0
----  ------------   ------ ------ ------  ----    ---  ------
fed6b  BLITHESDALE        49      0      0      0      0      0
```

Note that if you attempt to use the *pc_name* (**//blithesdale**), this command produces the following error:

```
$ probenet  -n //blithesdale
2AC00  unknown       ?(probenet) Unable to get information. -
too many transmit retries (OS/network)
```

uctnode — Uncatalog a PC (uncatalog_node)

Purpose

Removes the specified entry directory name from the local copy of the network root directory. This command does not accept as an argument the hexadecimal node ID *pc_node_id*. To uncatalog the PC, you must first catalog it with the **ctnode** command, described earlier in the chapter.

Format

uctnode [*node_name*]

Options

node_name	Specifies the entry node name of the PC.

Example

The following is the correct usage of the command:

```
$ uctnode blithesdale
```

You will get an error message if you attempt to uncatalog a PC using the *pc_node_id* (hexadecimal ID).

```
$ uctnode fed6b
```

```
?(uctnode)   Unable to uncatalog "fed6b" - name not found (OS/
naming server)
```

◆

Appendix A

Messages and Process Status Errors

This appendix first lists the messages that may appear on your PC's screen when you are using the Domain/PCI Connection, along with an analysis of each message and solution to the problem, if any. It then describes errors that may occur with the status of the server and DTERM processes.

(The "How DOS Responds to Permission Violations" section in Chapter 10 describes the error messages that may be displayed on your PC's screen in response to permission violations.)

Messages

Account manager error

You entered an invalid username. Use the **net login** command to re-enter a valid username.

Attempt to use local name

You tried to connect to a remote resource but incorrectly specified your computer's netname. With a version 1 or 2 Domain/PCI SIO connection product, the netname is always **domain**.

Can't open address file

When the Domain/PCI Connection tried to open the **netaddr** file, you received an error from DOS. Make sure that the **netaddr** file is present. The **netaddr** file is included on the Domain/PCI Connection master diskette.

Can't read transport status

The transport program cannot report status. Reboot your PC and try to start the Domain/PCI Connection again.

Cannot disconnect from current drive

You attempted to disconnect your current drive from a resource. Switch drives and try the command again.

Cannot send call packet to server

You attempted to make a Domain/PCI Connection without starting the server process. Start the Domain/PCI Connection on the server side before making the Domain/PCI Connection on the PC.

Command completed successfully

You issued a command that was successfully completed.

Command line option error

You typed an option incorrectly or used an option incorrectly. Check the syntax of the option within the manual or use **net help**.

Command operands incorrect

You typed a **net** command incorrectly. Check the syntax of the command by typing **net help** or refer to the manual.

Command parameters incorrect

You typed a **net** command incorrectly. Check the syntax of the command by typing **net help** or refer to the manual.

Connection refused

You used an invalid password when trying to connect to an offered resource, or you tried to connect to a resource that has not been offered.

Current drive is no longer valid

You used the **net disconnect** command (or **net logout**) from a drive you were connected to. Enter a connected (or physical) drive name.

Device is not redirected

You have specified a printer to which you are not connected. You must connect to a remote printer with the **net connect** or **use** command before you can use the remote printer.

Drive in use

You have specified in a **net connect** command a drive letter that is already being used. Check the list of drives in use by typing **net connect**.

DTERM is RESIDENT, USE HOTKEY

You attempted to invoke DTERM more than once. Toggle into DTERM with the Hot Key feature.

Duplicate redirection

You attempted to connect to a drive to which you already connected. Disconnect from the drive before attempting to reconnect.

Enter Password:

Appears when you use the **net login** command or the **-login** option. Enter your password.

Also, appears in Domain/PCI SIO connection products when you used an asterisk in a **net use** command to specify that the password should be typed in. Type in the password to continue operation.

Error writing to *<filename>*

An error occurred while writing to the specified file. The print command will not complete successfully. Check the remote computer and enter the command again.

File not found

The file you requested was not found. Use the DOS **dir** command to verify that the file exists and that you have typed its name correctly.

File not found:*<filename>*

The **net** command could not find the file required to complete the requested operation. Check that you have correctly created the required files.

Illegal address file

The **netaddr** file is missing or has been corrupted. Reinstall the Domain/PCI Connection.

Incorrect DOS version

You are trying to run the Domain/PCI Connection with an inappropriate version of DOS. You must be running DOS version 3.10 or higher.

Incorrect number of operands

You did not specify a keyword in the command. Check the command syntax by typing **net help** or consult the manual.

Invalid address string in netaddr file

The **netaddr** file contains an incorrect entry. Make a new copy of the **netaddr** file from the Domain/PCI Connection master diskette.

Invalid command line argument

An argument (or arguments) you entered was incorrect. Check the argument and re-enter.

Invalid device

You have mistyped a device name. Check the device name and try the command again.

Invalid device name

The device name you entered is not valid. Check the device name and re-enter.

Invalid drive specification

The drive you specified is not a legal drive. If you specified a Domain (virtual) drive, check your connections to drives with **net connect** to see if you've connected to the drive. Also, check the **config.sys** file to be sure that your **lastdrive** command includes the drive you've entered. Re-enter a legal drive specification.

Invalid drive was specified

You omitted a space between the drive you specified in your **net connect** command and the backslash character. Re-enter with a space between the drive and the backslash.

Invalid node id

The node id you entered is not a valid server id. Check the node ID and re-enter.

Invalid shortname or password

You have mistyped the pathname or password. Ask your system administrator for the list of resources and their passwords.

Invalid username or password

Your username or password was not correct. Re-enter the correct username and password at the appropriate prompts.

Missing password

You forgot to enter your password after the Password: prompt. Re-enter your password at the Password: prompt.

NET804: Network busy

The server node was too busy to respond to a request from your PC in the allotted response time. You need to increase the response time and then restart the server.

To increase the response time, use the **-retries** option to increase the number of retries allowed.

For Domain/PCI SIO connection products, specify the **-retries** option when you restart the server.

NET805: Network device no longer exists error reading drive *n* Abort, Retry, Ignore?

The connection between the server node and your PC has failed while the Domain/PCI Connection was accessing drive *n* on your PC.

First make sure that the cables are connected. If they are not, reconnect them and respond with **R** (for retry) to the **Abort, Retry, Ignore?** prompt.

If the cables are securely connected, you may have been switched out of the server due to heavy network traffic. Respond with **I** (for ignore) to the **Abort, Retry, Ignore?** prompt until the drive *n* prompt reappears. Then disconnect from drive *n*. Increase your **-retries** invocation option to decrease further occurrences of the problem.

You may need to restart the Domain/PCI Connection software on the server node.

Network adapter hardware error

You attempted to make a Domain/PCI Connection without starting the server process. Start the Domain/PCI Connection on the server side before making the Domain/PCI Connection on the PC.

Network already started

You tried to start the Domain/PCI Connection after it has already been started. To restart, reboot your PC.

Network error

An unexpected error was reported from the Domain system. Your command cannot be completed.

Network not installed

You tried to execute a **net** command without successfully reaching the target server. Restart the server and restart the Domain/PCI Connection before executing a **net** command.

Network not started

You tried to execute a **net** command before starting the Domain/PCI Connection. Start the Domain/PCI Connection with the **net start** command. Or, if you have merged the startup file with your **\autoexec.bat** file, start the Domain/PCI Connection by restarting DOS.

Network path not found

The pathname you used does not exist. Make sure that the pathname you used is correct.

No entries in redirector list

You typed the **net connect** command but you do not have any connections to Domain resources.

Node id required

You forgot to use the **–on** option or specified the **–on** option without specifying a node ID. Re–enter the command with a node ID after the **–on** option.

Password:

Enter your password at this prompt. The Password: prompt is a part of the **net login** command and the **–login** option.

Path not found

The pathname you entered is either incorrect or not accessible from the network.

Please log in:

Enter your username at this prompt. The login prompt is a part of the **net login** command and the **–login** option.

Please remove DTERM before logging out ***

You toggled out of DTERM and used **net logout** without first logging out of DTERM. Toggle back into DTERM and use the **logout** command to remove DTERM.

Remote server listen not posted

The server to which you attempted to connect or are connected to is currently not available. The system administrator should check to see if the server process is running. If not, restart the server.

Remote server not responding

The server to which you attempted to connect or are connected to is currently not available. The system administrator should check to see if the server process is running. If not, restart the server.

Too many arguments

You used too many arguments after a command line option. Check the format of the command in this manual.

Too many redirections

You've exceeded the maximum number of connections or servers allowed.

Unknown option

The option you entered is incorrect. Check the format of the command in this manual.

User already logged in

You attempted to log in more than once. Log in only once with a specific username and password per session.

User login required	You forgot the **net login** command or the **-login** option. You must log in before you can use the **net connect** command.
User name already specified	You attempted to enter your username more than once. Log in with a specific username only once per session.
Xport already installed	You attempted to restart the Domain/PCI Connection after a **net start** command had successfully executed.

Process Status Errors

Ordinarily, the server and DTERM processes receive manager status in the login protected subsystem during the Domain/PCI Connection installation procedure. (*Using Your Aegis Environment* describes protected subsystems.)

This section describes what happens when the server process, **server**, and DTERM process, **dterm_domain**, do not have manager status, and how to explicitly give them manager status.

Server Process Status

When the server process, **server**, does not have manager status in the login protected subsystem, you will be unable to copy or read Domain files from your PC, and the error log will contain an entry such as the following:

```
***************************************************************
Entry created on  01-Jul-1988 10:44     Node id: 245C
Process uid: 2EA1392C.245C

Setting virtual terminal process sid - no right to perform
operation (OS/ACL manager)
***************************************************************
```

To solve the problem, perform the following steps:

1. Verify that **server** does not have manager status by typing the following command:

   $ subs //*node*/sys/dpci/server

   where *node* is the name of the server node.

The following message indicates that **server** does not have manager status:

```
"//NODE/sys/dpci/server" is a nil subsystem manager
"//NODE/sys/dpci/server" is a file subsystem data
    object
```

2. Reassign manager status to **server** by logging in with a sys_admin account and typing the following:

```
$ ensubs login
$ subs //node/sys/dpci/server login -mgr
$ <CTRL> Z
```

3. Confirm that **server** now has manager status by typing the following:

```
$ subs //node/sys/dpci/server
```

The following message indicates that **server** has the proper manager status:

```
"//NODE/sys/dpci/server" is a LOGIN subsystem manager
"//NODE/sys/dpci/server" is a file subsystem data
    object
```

DTERM Process Status

When the DTERM process, **dterm_domain**, does not have manager status in the login protected subsystem, you will be unable to start DTERM, and the error log will contain a message such as "Unable to set SID."

To solve the problem, perform the following steps:

1. Verify that **dterm_domain** does not have manager status with this command:

```
$ subs //node/sys/dpci/dterm_domain
```

where *node* is the name of the server node.

The following message indicates that **dterm_domain** does not have manager status:

```
"//NODE/sys/dpci/dterm_domain" is a nil subsystem
   manager
"//NODE/sys/dpci/dterm_domain" is a file subsystem
   data object
```

2. Reassign manager status to **dterm_domain** by logging in with a sys_admin account, and typing the following:

```
$ ensubs login
$ subs /node/sys/dpci/dterm_domain login -mgr
$ <CTRL> Z
```

3. Confirm that **dterm_domain** now has manager status by typing the following:

```
$ subs //node/sys/dpci/dterm_domain
```

The following message indicates that **dterm_domain** has the proper manager status:

```
"//NODE/sys/dpci/dterm_domain" is a LOGIN subsystem
    manager
"//NODE/sys/dpci/dterm_domain" is a file subsystem
    data object
```

* * *

Appendix B

Command Summary

This appendix lists the commands that you use with the Domain/PCI Connection. The first commands in the list are commands that use keys, such as <ALT>; the rest are commands that you type at the DOS prompt. Note that this appendix *does not* list standard DOS commands. Refer to your DOS command manual for standard DOS commands. Refer also to the appropriate Domain/OS command reference for the environment you're using for Domain/OS commands.

<ALT> B	Enters the VT100's SET-UP B mode from SET-UP A mode when you are running DTERM.
<ALT> I	Starts the use of a script file in DTERM.
<ALT> O	Starts the creation of a script file in DTERM; pressing <ALT>O again stops the creation of the script file.
<ALT> <RIGHT SHIFT>	Executes the alternative DTERM Hot Key feature if you invoked DTERM with the **-alt** option.

<ALT> S	Enters the VT100's SET-UP A mode when you are running DTERM; press <ALT>S again to save any changes and exit from SET-UP A or SET-UP B mode.
<CTRL> D	Sends a mail message in DTERM if you are using the BSD or SysV environment.
<CTRL> S	Suspends the scrolling of information on the PC screen; press <CTRL> S again to resume scrolling. In DTERM set-up mode, saves changes.
<CTRL> Z	Sends a mail message in DTERM if you are using the Aegis environment.
<ESC>	In DTERM set-up mode, exits from the set-up mode without saving your changes.
<LEFT SHIFT> <RIGHT SHIFT>	Executes the Hot Key feature.
<SHIFT> <ESC>	Stops the DTERM program, leaving the server wit the access rights associated with your username Use the **logout** command to exit from DTERM an remove your permission rights from the server.
dpci*product server_id*	Invokes the startup file of a specified Domain/PC direct connection product (*product* as listed in Ta ble 1-1) on a specified server (*server_id*).
dpci1	Invokes the startup file of either of the Domain PCI SIO connection products.
dterm	Starts the DTERM program.
logout	Stops the DTERM program.
net connect	Connects your PC to Domain resources *without* se ting up an offers file for virtual drives.
	Lists the names of the Domain resources con nected to your PC's devices.

net continue	Resumes suspended connections to Domain resources.
net disconnect	Disconnects your PC from Domain resources.
net help	Provides information about the net commands.
net login	Logs you in to Domain/PCI and creates the initial server processes.
net logout	Logs you out of Domain/PCI, breaks all connections, and removes the server processes created when you logged in.
net offers	Lists the offers in the offers file.
net pause	Suspends connections to Domain resources.
net start	Enables you to start the network without making any connections.
net use	Connects (and disconnects with **/d** option) a PC device to a Domain resource. (Use only for version 1.0 and 2.0 Domain/PCI SIO connection products.)
netsvc	Enables you to disconnect your PC from the Domain Token Ring network services without disconnecting from your resources or without breaking connections. The **–n** option disables network services and the **–a** option enables network services.
pcinstl dpci*product drive*:	Installs the specified Domain/PCI Connection software on your PC's specified drive.
pci_assist	Examines a Domain directory and displays the DOS equivalent for names in that directory.

Appendix C

The Domain/PCI Connection Client Program Diskette

The Domain/PCI client program diskette contains the software component of the Domain/PCI Connection that runs on your PC. When you install the software, all the files (except **pcinstl.bat**, which installs the Domain/PCI Connection software) that are common to all Domain/PCI Connection products are copied to the **\dpci** directory. All files that are specific to the product are copied to the product-name specific directory, **\dpci***product* (where *product* is the product you're using, as listed in Table 1-1), below the **\dpci** directory. Thus you will get only those files in the **\dpci***product* directory that are specific to your installation.

The diskette is a standard 5-1/4-inch double-sided, double-density diskette. Keep this diskette in a safe place; it is your master diskette. Chapter 2 describes how to make a copy of the master diskette to use as a work diskette copy.

The client program diskette contains the following files:

Executable Files in \dpci

append.com	The append program, used with the **append** command.
connect.exe	The connect program, used with the **net connect** command.
\dpci*product*\xport.exe	The transport program, part of the Domain/PCI Connection that runs on your PC (where *product* is the product you're using as listed in Table 1-1).
\dpciring\netsvc.exe	The netsvc program, used with the **netsvc** command. This command enables your PC to disconnect from the Domain Token Ring network without disconnecting from resources or without breaking connections.
\dpciring\ringdiag.exe	The diagnostic program from the Domain/PCI–Ring board. This program gives information, such as the node ID of the ring adapter board.
dterm.exe	The DTERM program, used with the **dterm** commands.
login.exe	The login program, used with **net logout** and the **–login** option.
logout.exe	The logout program, used with the **net logout** command.
net.exe	The net program, used with the **net** commands.
prtsc.exe	The program that implements the <CTRL> <ALT> <PrtSc>.
redir.exe	The redirector program, used to determine whether requests are coming from a local or remote source.
session.exe	The session program Domain/PCI Connection that runs on your PC.
sessionr.exe	The Domain/PCI session program Domain/PCI Connection that runs on your PC.
setname.exe	The program that implements the **net name** command. Used internally by the Domain/PCI Connection.
typehelp.exe	The program that displays the help files.
use.exe	The program that implements the **net use** command.

Help Files in \dpci

connect.hlp	The help file for the **net connect** command.
login.hlp	The help file for the **net login** command.
logout.hlp	The help file for the **net logout** command.
net.hlp	The help file for all the **net use** commands.
pause.hlp	The help file for the **net pause** and **net continue** commands.
start.hlp	The help file for the **net start** command.
use.hlp	The help file for the **net use** command.

Additional Files in \dpci

dpci1.bat	The client startup file that starts the Domain/PCI SIO connection products.
dpci*product*.bat	The client startup file that starts the appropriate Domain/PCI direct connection product on PCs (where *product* is the product you're using as listed in Table 1-1).
msnet.ini	The control file for the **net.exe** file.
netaddr	The file used by **session.exe** to determine the address of your PC and the address of the server node.
readme.doc	The file that describes the files provided with the Domain/PCI Connection.

Files in Top-Level or Root Directory

pcinstl.bat	The batch file that installs the Domain/PCI Connection software.

Files on Domain/PCI-Enet Diagnostic Software Diskette

enetdiag.exe	The diagnostic program for the Domain/PCI-Enet Ethernet board. This program obtains information, such as the node ID of the Ethernet board. Note that this program is found on the second Domain/PCI-Enet diskette, the enetdiag diskette.

Appendix D

Customizing the Client Startup and \autoexec.bat Files

This appendix describes how to customize two files that will enable you to log in and automatically make customized connections when you turn on your PC. These files are called

- Domain/PCI client startup file
- **\autoexec.bat** file

Both files are presented below.

Customizing the Domain/PCI Client Startup File

You can customize the basic commands included in the default client startup file to meet your particular needs. The default Domain/PCI client startup file is located in **\dpci\dpci*product*.bat** where *product* is the name of the product you're using.

You can use a text editor, such as **edlin** to edit the file. The following is a sample annotated Domain/PCI client startup file:

```
: *** DPCI-product Client Startup File ***
: ***
.
.
[1] net start dpciproduct %2 %3 %4 %5 %6 %7 %8
[2] net login -on %1
[3] : ***Insert your NET USE or NET CONNECT commands here***
[4] net connect d: //
[5] net connect lpt2: print
.
.
```

When you have completed customizing your client startup file, you can invoke it from your **\autoexec.bat** file. Refer to "Customizing Your PC's \autoexec.bat File" later in this appendix for more information on how to merge the batch files.

NOTE: Once you've completed customizing your startup file, you might want to rename the file and invoke it from the **\autoexec.bat** file using the new name. Renaming the file ensures that your customized client startup file will not be overwritten if you reinstall the Domain/PCI connection software on your PC.

The following sections describe each important line in the client startup file:

- Editing the **net start** command
- Editing the **net login** command
- Editing the **net connect** commands

Editing the net start Command

Line [1] of the client startup file is the **net start** command, which installs and starts some Domain/PCI programs and files, but does not actually make any connections to the Domain system.

To start the Domain/PCI Connection, the client startup file uses the following command:

```
net start dpciproduct %2 %3 %4 %5 %6 %7 %8
```

where *product* is the product name and the symbols %2 %3 %4 %5 %6 %7 %8 are the variables that are replaced by invocation options you entered in your invocation command, described in Chapter 2.

You can add options to the **net start** command to customize the startup file. The options are described in the following section "net start Options."

net start Options

The Domain/PCI direct connection products have one **net start** command option. The Domain/PCI SIO connection products have three **net start** command options. You can include one or more options to customize the command.

Direct Connection Products Options

–retries *zz*

Specifies the number of times the PC should try to execute a command again after encountering an error. The default number is 60. The maximum number of retries is 250. If you are on a busy network, you may want to use a number greater than the default to ensure that you stay connected to the network.

For example,

```
net start dpciring -retries 95
```

SIO Connection Products Options

–line *n*

Indicates the number of the asynchronous communication adapter connector where the cable for the Domain/PCI Connection server is plugged in; this number *n* can be 1 or 2. **1** indicates COM1 (PRIMARY or LINE 1) and **2** indicates COM2 (ALTERNATE or LINE 2).

For example,

```
net start dpcil -line 1
```

–baud *yyyy*

Specifies the baud rate, where *yyyy* is the rate. The value of *yyyy* can be 300, 600, 1200, 2400, or 4800.

For example,

```
net start dpcil -baud 4800
```

You need to specify this option only if you have established a baud rate on the server node that is different from 9600. (System administrators: see Chapter 9.)

–retries *zz*

Specifies the number of times the PC should try to re–execute a command after encountering an error. The maximum number of retries is 250. The default number is 10. If you are on a busy network, you may want to use a number greater than the default to ensure that you stay connected to the network.

For example,

```
net start dpcil -retries 25
```

Editing the net login Command

Line [2] of the client startup file is the **net login** command, which prompts you to log in. For Domain/PCI direct connection products, the **net login** command also establishes which initial server node you're connecting to.

To log into a Domain/PCI Connection direct connection product, the client startup file uses the following command:

```
net login -on %1
```

where the symbol %1 is a variable that uses the *server_id* (hexadecimal node ID) you entered in your **dpci***product* invocation command, described in Chapter 2.

You can customize this command by adding your username and by replacing the %1 with the actual *server_id* of the server you will be using in your Domain/PCI Connection. For example:

```
net login beth -on 18dfc
```

To log into a Domain/PCI SIO connection product, your client startup file uses the command shown above without the **–on** option. You can add your username to the command. For example:

```
net login beth
```

Editing the net connect Commands

Line [3] of the Domain/PCI client startup file contains the following comment:

```
: ***Insert your NET CONNECT commands here***
```

You can further customize your client startup file by adding any special **net connect** or **net use** to this section of the client startup file. Chapter 5 provides descriptions of the commands you can add to this section. Keep in mind that you can make up to 10 connections to different directories.

Lines [4] and [5] of the Domain/PCI client startup file contain two **net connect** commands. If necessary, these commands can be customized to specify connections to specific Domain directories.

Line [4] is the first **net connect** command. This command enables you to connect your PC to Domain resources without setting up an offers file for virtual drives. Chapter 4 contains information about offers files. The following line shows the first **net connect** command line in the Domain/PCI client startup file.

```
net connect d: //
```

This command establishes your PC drive D as the the top level or root of the Domain network. This command should not be edited or removed.

You can add another command line to connect to another drive and/or to another directory. For example, the command line shown below, connects you to a specific directory when you use drive E.

```
net connect e: //corey/beth
```

Line [5] is the second **net connect** command. This command enables you to associate the name of a PC printer device with a Domain printer. The command in the Domain/PCI client startup file connects your PC's printer device, **lpt2**, to the default Domain printer called **print** in the offers file.

```
net connect lpt2: print
```

You can customize this line to use a different printer device and/or to connect to another printer specified in your offers file. However, you can connect a printer device to only one printer at a time.

Customizing Your PC's \autoexec.bat File

You can add a command to your PC's **\autoexec.bat** file so that it invokes the Domain/PCI Connection when you power up your PC. Specifically, this command will automatically invoke the Domain/PCI client startup file, which we described above. This section shows you how to customize your **\autoexec.bat** file so that it merges with the client startup file.

To customize your **\autoexec.bat** file, do the following:

1. Append **c:\dpci** to your **path** command. This will ensure that you can execute the net commands or the default batch files from any directory.

2. Add the invocation command **dpci***product server_id* (for Domain/PCI direct connection products) or **dpci1** (for Domain/PCI SIO connection products), which invokes your Domain/PCI client startup file. This must be the last command in the batch file. The invocation command invokes the appropriate Domain/PCI client startup file.

For example, suppose your **\autoexec.bat** file contains

```
.
.
prompt $p$g
path c:\dos3.10
```

You can modify it for Domain/PCI-Ring by adding

```
.
.
prompt $p$g
path c:\dos3.10;c:\dpci;
dpciring 18dfc
```

where **dpciring** is the name of product you're using and **18dfc** is the hexadecimal node ID of the server node you want to connect to.

Note that **c:\dpci** is appended to the path and that **dpci***product* is the last command in the file. (See Table 1-1 for the values of *product*.) Refer to your DOS manual for more information about **\autoexec.bat**.

◆

Appendix E

Updating the Client Software from an Install Area

Once the user has installed the Domain/PCI Connection client software on the PC using the procedures described in Chapter 2, the system administrator can create an **install area** on the Domain system. An install area is a Domain directory that contains all the installation software for a product. From this directory, the PC user can update the client software on any PC on the network.

This appendix tells the system administrator how to set up the install area and tells the PC user how to update the client software.

Setting Up the Install Area

This procedure copies the client software from a PC to the install directory on the Domain system. The procedure assumes that the Domain/PCI Connection software has already been installed at least once on the PC according to the procedures described in Chapter 2.

Use the following procedure to create an install area:

1. From the server node, create a directory into which you will copy the new Domain/PCI Connection client software. For example, **//admin/install/dpci/pcfiles**.

2. From a PC, start the Domain/PCI Connection according to the procedure "Starting the Domain/PCI Connection" in Chapter 2.

3. Connect to the install directory on the server node. This is the directory that you created in Step 1. For example,

   ```
   C:> net connect e:  //admin/install/dpci/pcfiles
   ```

4. Insert a new Domain/PCI Connection client diskette in drive A.

5. Change the current drive to A: and press <RET>.

   ```
   C:> A:
   ```

6. At the DOS prompt, type the following command and press <RET>:

   ```
   A:> pcinstl all drive:
   ```

   where *drive* is the name of the drive that you connected to in Step 3.

   This command invokes a batch file that copies the new Domain/PCI Connection client software to your install area. From this directory, PC users can update their client software by following the procedure described in the next section.

7. Follow the messages on the screen.

Updating the Client Software

This procedure updates the client software on the PC from the install directory on the Domain system. The procedure assumes that the Domain/PCI Connection software has already been installed at least once on each target PC according to the procedures described in Chapter 2.

1. From a PC, start the Domain/PCI Connection according to the procedure "Starting the Domain/PCI Connection" in Chapter 2.

2. Connect to the install directory on the server node. This is the directory that the system administrator created in Step 1 of "Setting up the Install Area." For example,

   ```
   C:> net connect e: //admin/install/dpci/pcfiles
   ```

3. Change the current drive to E: and press <RET>.

   ```
   C:> E:
   ```

4. At the DOS prompt, type the following command and press <RET>:

   ```
   E:> pcinstl dpciproduct drive:
   ```

   where *product* is the name of the product software you want to update (as listed in Table 1-1) and *drive* is the name of the drive on which your client software is found. You can optionally choose **all** in place of **dpci***product* if you want to update the client software for all of the Domain/PCI Connection products.

   This command invokes a batch file that updates the existing Domain/PCI Connection client software with new software.

5. Follow the messages on the screen.

Index

Symbols are listed at the beginning of the index. Entries in color indicate task-oriented information.

Symbols

(comment character), 9–13

; (comment character), 4–7

> (directive character), 4–7

// (double slashes), 3–5

*** (null character), 4–12

\ (backslash), 3–13

/ (slash), 3–13

A

access control list (ACL), 10–1
  See also permission rights

accessing files, 4–17 to 4–18
  created on Domain from your PC, 4–12 to 4–15
  "invisible" files, 7–6

accessing protected files and directories. *See* permission rights

account protection, 6–5

adding comment lines to the file type conversion file, 4–16 to 4–17

adding connections, 3–6 to 3–7

Aegis environment
  shell commands, 1–6, 7–2

–alt option, (DTERM) 6–2

alternative hot key feature (DTERM), 6–10

append (DOS command), 4–2 to 4–4

application
  invoking a VT100, 8–2
  PC, storing on Domain, 4–10 to 4–11
  word processing, using, 4–18

assembler files, Domain 4–15

assign (DOS command), 4–2, 4–4 to 4–5

asynchronous communication adapter connector, 2–11

\autoexec.bat file
  customizing your PC's, D-7 to D-8
  description of, 2-13

B

backup (DOS command), 4-15 to 4-16

-baud
  invocation option, 2-11
  net start option, 5-29, D-4
  server invocation option, 9-14

baud rate, 2-12

bldt (Domain/OS command), 2-10, 11-2 to 11-3

\block argument, 9-9

BSD environment
  shell commands, 1-6, 7-2

bytes free, 3-5

C

changing the application's default printer, 4-27

\char argument, 9-5

chkdsk (DOS command), 3-6

client software
  definition of, 2-1
  installing, 2-3 to 2-9
  program, 2-3
  updating, E-1 to E-4

client startup file, customizing, D-1 to D-7

COM1, 2-11

COM2, 2-11

command summary, B-1 to B-3

commands, net. *See* net commands

  2-10, 11-2 to 11-3
  ctnode, 11-4
  lcnode, 11-5
  netstat, 11-6 to 11-7
  probenet, 11-8 to 11-9
  uctnode, 11-10

comment lines, adding to the file type conversion file, 4-16 to 4-17

compiler files, 4-15

\config.sys, 3-7 to 3-8

configuration file, server, 9-11

configuring the server, 9-11 to 9-13

connect
  additional drives 3-6 to 3-7
  to a Domain printer, 5-9 to 5-10, 5-33 to 5-34
  drives to Domain directories, 5-5 to 5-8, 5-30 to 5-32
  initial drive, 3-3 to 3-6

converting files without extensions in the conversion file, 4-12

copy (DOS command), 4-19, 4-23 to 4-24

copying files, 4-19

cps (Domain/OS command), 9-13

creating files on Domain from your PC, 4-9 to 4-10

cross compiler files, Domain, 4-15

ctnode (Domain/OS command), 11-4

customizing the client startup file, D-1 to D-7

D

Danford SEU board, 9-15

default client startup file
  See also client startup file
  invoking the, 2-9 to 2-11

default printer, 3-1

dir (DOS command), 3-5

direct connection product, 1-5

directives, 4-7

directory
  case differences when connecting, 3-12
  displaying contents of, 3-5
  displaying in DTERM, 7-4
  removing, 3-15
  spool, 10-4

disconnect
  from a Domain directory, 5-15 to 5-16, 5-35
  from a Domain printer, 5-17, 5-36

disk free space, 3-5

diskette
  master, 2-4
  work. *See* work diskette

diskette drives, installing client software on, 2-6 to 2-9

diskcomp (DOS command), 3-6

diskcopy (DOS command), 3-6

display print queue, 4-27

Display Manager (DM), 6-1

displaying Domain/OS filenames from DOS, 3-13 to 3-15

displaying filenames in DTERM, 7-4

displaying help information, 5-18 to 5-19

displaying search paths, 4-4

DM (Display Manager), 6-1

Domain. *See* Domain system and specific entries

Domain/Access files, 4-13, 4-14

Domain assembler files, 4-15

Domain cross compiler files, 4-15

Domain default printer, 3-1

Domain NFS files, 4-13 to 4-14

Domain/PCI Connection
  client program diskette, C-1 to C-4
  convention in manual, 1-5
  definition of, 1-1
  installing (client software), 2-3 to 2-9
  introduction to, 1-1 to 1-6
  making the initial, 3-1 to 3-16
  managing the server, 9-1 to 9-19
  product family, 1-2 to 1-4
  starting, 2-9 to 2-12

Domain/PCI-1, 1-3

Domain/PCI-8
  definition of, 1-3
  distribution panel, 1-3

Domain/PCI-Enet, 1-2

Domain/PCI-Ring, 1-3

Domain resources

- See also resources
- definition of, 1-6

- reconnecting to, 3-16
- resume connections to, 5-13 to 5-14

Domain system
- connection 1-2
- and DTERM, 7-1 to 7-9
- files, differences between, 3-9 to 3-13
- root directory, 2-9
- top-level directory, 2-9
- uasc files, 4-15
- using printers, 4-25 to 4-26

DOS
- commands use with Domain/PCI, 3-6
- file format (8.3 character format), 3-13
- files, difference between, 3-9 to 3-13
- prompt, 3-4
- software, 2-6

DPSS/Mail, 1-6, 6-1, 7-3 to 7-4

drives
- fixed, 2-5 to 2-6
- logical, 2-3 to 2-4
- physical, 3-8
- specifying an additional number of, 3-7
- virtual, 1-2, 3-8

DTERM
- *See also* VT100 terminal emulation
- definition of, 1-6
- display, 6-6
- getting started with, 6-1 to 6-13
- graphic applications, 6-2
- process status, A-14 to A-15
- starting, 6-2 to 6-4
- starting up with a specific program, 6-11 to 6-13
- status line, 6-7 to 6-9
- stopping, 6-10 to 6-11
- using with Domain system, 7-1 to 7-9

E

editing
- net commands, D-6 to D-7
- text files. *See* **edlin** text editor

edlin (DOS command) text editor, 3-8

error log file, 10-4 to 10-6

extensions, 4-7

F

fdisk (DOS command), 3-6

file type conversion file
- adding comment lines, 4-16 to 4-17
- converting files without extensions, 4-12
- definition of, 4-6 to 4-7
- initial, 4-7 to 4-8
- locating, 4-9, 4-12
- **pci_cvrt.dat** file, 4-6
- sharing files with Domain users, 4-6 to 4-17

filename
- display differences, 3-11
- displaying Domain/OS, from DOS, 3-13 to 3-15
- displaying, in DTERM, 7-4
- extensions, 3-11

files
- accessing, 4-17 to 4-18
- accessing, created on Domain from your PC, 4-12 to 4-15
- accessing "invisible," 7-6
- copying, 4-19
- creating, on Domain from your PC, 4-9 to 4-10
- graphics, printing, 4-21
- manipulating, 4-1 to 4-28
- naming conventions for, 3-9
- print queue, 4-27
- printing, 4-20 to 4-27
- protected, and directories. *See* permission rights
- reading, 3-5
- script, 7-7 to 7-9
- scrolling through, 3-5
- sharing files with Domain users, 4-6 to 4-17
- text. *See* text file

fixed disk, installing client software on, 2-5 to 2-6

-font option, 9-5

format (DOS command), 3-6

G

-gr option (DTERM), 6-2

graphics applications, 6-2

graphics character set support on DTERM, 8-3

Graphics printer, IBM, 4-20

graphics, printing files with, 9-7

H

hard disk. *See* fixed disk

help, displaying information about the commands, 5-18 to 5-19

hexadecimal server ID, 2-10

hot key feature (DTERM), 6-9 to 6-10

I

IBM Graphics printer, 4-20

ID, server, 2-10

increasing the number of virtual drives, 3-8

initial connection, 3-3 to 3-6

initial file type conversion file, 4-7 to 4-8

install area
- definition of, E-1
- setting up the initial, E-2 to E-3

installing the client software
- on diskette drives only, 2-6 to 2-9
- on a fixed disk, 2-5 to 2-6

installing the server software, 2-2

interconnection solutions, 1-2

"invisible" files, 7-6

invocation options, 2-11 to 2-12

- invoking a VT100 application, 8-2

J

- **join** (DOS command), 3-6

K

- key terms, 1-5
- **kill** (Domain/OS command), 9-18

L

- **label** (DOS command), 3-6
- **lastdrive** command, 3-7 to 3-8
- **lcnode** (Domain/OS command), 11-5
- **-line**
  - invocation option, 2-11
  - net start option, 5-29, D-4
  - server invocation option, 9-14
- list PC connections, 5-11 to 5-12, 5-37 to 5-38
- listing files that don't display in print queue, 4-27
- local area network, 1-2
- locating the file type conversion file, 4-9
- locked objects, 10-6 to 10-7
- logging in, 3-2 to 3-3, 5-20 to 5-21
  - to DTERM, 6-4
- logging out, 3-15 to 3-16, 5-22 to 5-23
  - of DTERM, 6-10
- logical connection, 2-3 to 2-4
- login account, 3-2
- **-login** option (DTERM), 6-3, 6-4
- **-login** option (net connect), 5-6
- **logout** command (DTERM), 6-10
- **lpt1**, 4-23
- **lpt2**, 2-9, 4-23
- **lpt3**, 4-23

M

- mail (DPSS/Mail), 1-6, 6-1, 7-3 to 7-4
- mailbox file, 10-4
- making additional connections, 3-6 to 3-7
- making an initial connection, 3-3 to 3-6
- managing the server, 9-1 to 9-19
- manipulating files, 4-1 to 4-28
- master diskette, 2-4
- **maxclients** command, 9-12 to 9-13
- maximum connections at one time, 9-12 to 9-13
- messages, A-1 to A-12
- modem, 2-9, 2-12
- MS-DOS software, 2-6
- multiple printer format, 9-8

N

- naming conventions for files, 3-9
- naming directories, 6-5
- net commands
  - descriptions of, 5-1 to 5-38
  - by task, 5-2 to 5-3
- **net connect** command
  - connect to a Domain directory, 5-5 to 5-8
  - connect to a Domain printer, 5-9 to 5-10
  - editing, D-6 to D-7
  - making an initial connection with, 3-3
  - list connections, 3-7, 5-11 to 5-12
  - and offers file, 4-20 to 4-21
  - using, instead of **net use**, 9-11
- **net continue** command, 5-13 to 5-14
- **net disconnect** command
  - disconnect from a Domain directory, 5-15 to 5-16
  - disconnect from a Domain printer, 5-17
- **net help** command 5-18 to 5-19
- **net login** command, 5-20 to 5-21, 10-2, D-5
- **net logout** command, 3-15, 5-22 to 5-23
- **net offers** command, 4-25, 5-24 to 5-25
- **net pause** command, 5-26 to 5-27
- **net start** command, 5-28 to 5-29, D-3 to D-5
- **net use** command
  - connect to a Domain directory, 5-30 to 5-32
  - connect to a Domain printer, 5-33 to 5-34
  - disconnect from a Domain directory, 5-35
  - disconnect from a Domain printer, 5-36
  - list connections, 5-37 to 5-38
  - using **net connect** instead of, 9-11
- **netstat** (Domain/OS command), 11-6 to 11-7
- NFS files. *See* Domain/NFS files
- node
  - definition of, 1-2
  - server, 1-2
  - server ID, 2-10
- **'node_data** directory, 10-3 to 10-4
- **\notranslate** option, 9-6

O

- offer definitions, 9-3
  - *See also* offers
- offers
  - creating, 9-10
  - directory, setting up, 9-9
  - editing, 9-10
  - initial, 9-10

- list the, 5–24 to 5–25
- name, 9–4
- printer, setting up, 9–4 to 9–8

- offers file
  - definition of, 4–20, 9–3
  - how it works, 4–20 to 4–21
- **-on** option (DTERM), 6–2, 6–4
- **-on** option (net connect), 5–6, 5–10

P

- pathname translation, 3–13
- pathnames, using, 7–3
- PC (Personal Computer)
  - definition of, 1–1
  - diskette drives, installing client software on, 2–6 to 2–9
  - equivalent for keyboard keys in DTERM, 8–4
  - fixed disk, installing client software on, 2–5 to 2–6
- PC-DOS. *See* DOS
- **pci_assist**, 7–5 to 7–6
- **pci_cfg.dat**, 9–11
- **pci_cmd.dat**, 4–20, 9–3
  - *See also* offers file
- **pci_cvrt.dat** file, 4–6
  - *See also* file type conversion file
- **-pcprint** option, 9–5
- permission rights
  - locked objects, 10–6 to 10–7
  - logging in and, 3–2
  - using, 10–1 to 10–8
- permission violations, 10–7 to 10–8
- physical SIO ports, maximum, 9–14, 9–15
- **-pingtime** option, 9–13
- ports, SIO
  - logical, 9–15
  - physical, maximum, 9–14, 9–15
- PostScript format, 9–8
- **-pr** argument, 9–4
- *prf_option*, 9–5
- **print** (DOS command), 4–22 to 4–23
- print queue, 4–27
- printer
  - changing the application's default, 4–27
  - default Domain, 3–1
  - IBM Graphics printer, 4–20
  - lpt2 connects to, 3–1
  - multiple printer format, 9–8
  - offers, setting up, 9–4 to 9–8
  - using Domain, 4–25 to 4–26
- *printerdevice* 4–23
- printing
  - from application programs, 4–26 to 4–27
  - files, 4–20 to 4–27
  - files that contain graphics, 9–7
  - files in PostScript format, 9–8
  - graphics from an application, 4–27
  - graphics files, 4–21
  - in multiple printer format, 9–8
  - text files, 4–21, 9–6
  - text files without file type conversion, 9–6 to 9–7
- **prn**, 4–23
- **probenet** (Domain/OS command), 11–8 to 11–9
- process status errors
  - DTERM process status, A–14 to A–15
  - server process status, A–13 to A–14
- *product* variable (table), 1–5
- **-program** option (DTERM), 6–2
- **prompt** (DOS command), 3–4
- prompt, Domain/OS, 6–5
- **ps -aux** (Domain/OS command), 9–16
- **pst** (Domain/OS command), 9–16

R

- reading a file, 3–5
- reconnecting to Domain resources, 3–16
- **recover** (DOS command), 3–6
- **\reset** option, 9–6
- resources
  - *See also* Domain resources
  - local, 1–6
  - remote, 1–6, 5–1
- responsibilities
  - of system administrator, 2–2
  - of user, 2–2
- **restore** (DOS command), 4–15 to 4–16
- resume connections to Domain resources, 5–13 to 5–14
- **-retries**
  - invocation option, 2–11, 2–12
  - net start option, 5–28, 5–29, D–4, D–5
  - server invocation option, 9–13, 9–14
- root directory, 2–9
- running graphic applications with DTERM in background, 6–2

S

- script files
  - creating, 7–7 to 7–8
  - using, 7–8 to 7–9
- scrolling through a file, 3–5
- search paths, displaying, 4–4
- *server_id*, 2–10
- server
  - configuration file, 9–11
  - configuring, 9–11 to 9–13
  - managing, 9–1 to 9–19
  - stopping, 9–18
  - Subject Identifier (SID), 6–3, 10–2
- server node
  - *See also* server
  - crash, 10–6
  - definition of, 1–2
- server process
  - Domain/PCI creates, 9–16 to 9–17

- maximum, 10-1
- number of user processes created, 9-17

- status, A-13 to A-14
- unconnected, 9-17

server software
  - definition of, 2-1
  - installation of, 2-2
  - program, 2-3

setting screen characteristics in DTERM, 8-5

\setup option, 9-6

set-up mode A, 8-5 to 8-7

set-up mode B
  - cursor type, 8-9
  - local echo, 8-9
  - margin bell, 8-9
  - using, 8-8 to 8-10
  - wraparound, 8-9

SEUs (Serial Expansion Units), 9-15 to 9-16

sharing files with Domain users, 4-6 to 4-17
  - *See also* file type conversion file

shell command
  - Aegis, 1-6, 7-2
  - BSD, 1-6, 7-2
  - Domain/OS, for system administration, 11-1
  - with DTERM, using, 7-1 to 7-2
  - SysV, 1-6, 7-2

SID (Server's Subject Identifier), 6-3, 10-2

sigp (Domain/OS command), 9-18

SIO connection product, 1-5

SIO ports
  - logical, 9-15
  - physical, maximum number of, 9-14, 9-15

spool directory, 10-4

start the network without connections, 5-28 to 5-29

starting the client software, 2-9 to 2-12

starting DTERM, 6-2 to 6-4

starting DTERM with a specific program, 6-11 to 6-13

starting a shell from Domain/OS, 7-2

startup file, D-1 to D-7

startup scripts, 6-5

stopping DTERM, 6-10 to 6-11

stopping the server, 9-18

storing a PC application on Domain, 4-10 to 4-11

subst (DOS command), 3-6

suspend connections to Domain resources, 5-26 to 5-27

sys (DOS command), 3-6

system administration command 11-1 to 11-10

system administrator
  - responsibilities of, 2-2
  - what the system administrat should read, 2-3, 9-2

SysV environment
  - shell commands, 1-6, 7-2

T

terms, key, 1-5

text editor, 3-8, 4-17 to 4-18

text file
  - definition of a, 4-6
  - printing a, 4-21, 9-6 to 9-7

top-level directory, 2-9

-transparent option, 9-5

tutorial, 3-1 to 3-16

type (DOS command), 3-5, 4-18

U

uasc file format, 4-6

uctnode (Domain/OS command), 11-10

uid directives, 4-14

UNIX system, 7-2

updating the client software. *See* install area

user, responsibilities of, 2-2

user_data directory, 4-12

using Domain printers, 4-25 to 4-26

using DTERM with the Domain system, 7-1 to 7-9

using **net connect** instead of **net use**, 9-11

using set-up Mode B, 8-8 to 8-10

V

VAX VMS systems, 4-14

ver (DOS command), 2-10

vi editor, 8-2

virtual drive
  - definition of, 1-2
  - increasing the number of, 3-8

VT100 applications
  - invoking, 8-2
  - using, with DTERM, 8-1 to 8-10

VT100 terminal emulation
  - *See also* DTERM
  - character attribute support, 8-2
  - definition of, 1-6
  - graphics character set support, 8-3
  - PC equivalent for keyboard keys, 8-4
  - setting screen characteristics, 8-5
  - using set-up Mode A, 8-5 to 8-7
  - using set-up Mode B, 8-8 to 8-7

volume label, 3-5

W

word processing application, using, 4-18

work diskette

copy client software, 2-8 to
  2-9
creating, 2-7
definition of, 2-6

◆